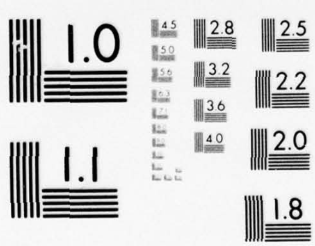
AD-A038 843 CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER --ETC F/G 15/4
CHANGE DETECTION AND ANALYSIS IN MULTI-SPECTRAL IMAGES,(U)
DEC 77 K E PRICE F44620-73-C-0074
UNCLASSIFIED AFOSR-TR-77-0448 NL

1 OF 3
AD
A038843

OF 3

38843



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

8

Change Detection and Analysis in Multi-Spectral Images

Keith Edward Price

December 18, 1976

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is
approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

7. AUTHOR(s) 10 Keith Edward Price	8. CONTRACT OR GRANT NUMBER(s) 15 F44620-73-C-0074
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Computer Science Dept. Pittsburgh, PA 15213 12 245p.	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 17 16 2304 A2
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd Arlington, VA 22209 11	12. REPORT DATE 18 Dec[redacted]977 13. NUMBER OF PAGES 102
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Air Force Office of Sciencific Research (NM) Bolling AFB, DC 20332	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

feature analysis to generate the symbolic description of the regions and image, use of knowledge to guide the segmentation and symbolic registration procedures, and lastly change analysis itself. ~~We present~~ several diverse scenes (house, cityscape, satellite images, aerial images, and radar images) are presented, each of which has a task description and a predefined set of knowledge elements, and will show how several different tasks can be performed with a general change analysis system.

Early segmentation techniques were either designed for specific applications or were very expensive. The segmentation of an image into regions by a histogram based region splitting procedure has proved to be useful over a wide range of images, but also tends to be expensive (Ohlander, 1975). In order to make this procedure more efficient, ~~we have incorporated~~ the use of "planning" was incorporated into the segmentation processing. This use of planning means that the segmentation is generated in about a tenth (or better) of the time required without planning. This segmentation method was originally developed for use on color (i.e. multi-spectral) images, but many of the images which we must analyze are monochromatic. We will present several alterations to the general segmentation method to use it on a wider range of scenes, including monochromatic images. The primary alterations are the addition of a few specific textural measures to aid in the segmentation of regions with certain textural properties, and use of special heuristics in the segmentation process so that partial segmentations are possible for the monochromatic images.

We present a set of features which can be used for symbolic description, matching, and change analysis. The features are grouped into classes of features similar to those used in human image understanding. These classes include: size, shape, color, position, etc. The set of features is by no means complete, but the addition of new features is straight forward.

The feature based descriptors of regions in two images of the same scene are used by the symbolic registration procedure to identify corresponding regions in the two images. The matching procedures uses a feature based distance metric to find the region in one image which corresponds to a region in another image (symbolic registration). For stereo pair analysis and symbolic matching tasks this is sufficient since only symbolic registration is required. For change detection tasks, further processing is required to generate the change information.

The tasks presented here range from a simple symbolic registration task to a complex task of the computation of the change in the number of occurrences of a particular type of region (i.e. the number of occurrences of a particular object). We present the results of symbolic registration for the six scenes. The results are not perfect, but most of the matching errors are traceable to the initial segmentation of the image. We also present alterations to a general segmentation method which reduces the time required for segmentation with this method by about a factor of 10. Several other alterations are given so that this procedure can be used effectively with monochromatic images. We present a method for symbolic change analysis which solves many of the problems encountered by signal based change analysis systems. The problems include the analysis of images with changes in the point of view of the observer, analysis of multi-spectral images without a corresponding increase in the complexity of the analysis, and effective analysis of the detected changes in the image.

Change Detection and Analysis in Multi-Spectral Images

Keith Edward Price

December 18, 1976

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA, 15213

Submitted to Carnegie-Mellon University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science.

This work is supported in part by the Defense Advanced Research Projects Agency under contract number F44620-73-C-0074 and is monitored by the Air Force Office of Scientific Research.

Change Detection and Analysis in Multi-Spectral Images

Keith Price

This thesis describes research toward the development of a general image understanding system. Our system has been directed toward the problem of the comparison of pairs of different images of the same scene to generate descriptions of the changes in the scene. Unlike earlier work in the change analysis area, we have performed all the matching and change analysis at a symbolic level rather than a signal level. To facilitate this symbolic analysis over a wide variety of images, advances in several other areas of image analysis were also required. These areas are: segmentation techniques to generate the basic units used in the symbolic analysis, feature analysis to generate the symbolic description of the regions and image, use of knowledge to guide the segmentation and symbolic registration procedures, and lastly change analysis itself. We present several diverse scenes (house, cityscape, satellite images, aerial images, and radar images), each of which has a task description and a predefined set of knowledge elements, and will show how several different tasks can be performed with a general change analysis system.

Early segmentation techniques were either designed for specific applications or were very expensive. The segmentation of an image into regions by a histogram based region splitting procedure has proved to be useful over a wide range of images, but also tends to be expensive (Ohlander, 1975). In order to make this procedure more efficient, we have incorporated the use of "planning" into the segmentation processing. This use of planning means that the segmentation is generated in about a tenth (or better) of the time required without planning. This segmentation method was originally developed for use on color (i.e. multi-spectral) images, but many of the images which we must analyze are monochromatic. We will present several alterations to the general segmentation method to use it on a wider range of scenes, including monochromatic images. The primary alterations are the addition of a few specific textural measures to aid in the segmentation of regions with certain textural properties, and use of special heuristics in the segmentation process so that partial segmentations are possible for the monochromatic images.

We present a set of features which can be used for symbolic description, matching, and change analysis. The features are grouped into classes of features similar to those used in human image understanding. These classes include: size, shape, color, position, etc. The set of features is by no means complete, but the addition of new features is straight forward.

The feature based descriptors of regions in two images of the same scene are used by the symbolic registration procedure to identify corresponding regions in the two images. The matching procedures uses a feature based distance metric to find the region in one image which corresponds to a region in another image (symbolic registration). For stereo pair analysis and symbolic matching tasks this is sufficient since only symbolic registration is required. For change detection tasks, further processing is required to generate the change information.

The tasks presented here range from a simple symbolic registration task to a complex task of the computation of the change in the number of occurrences of a

particular type of region (i.e. the number of occurrences of a particular object). We present the results of symbolic registration for the six scenes. The results are not perfect, but most of the matching errors are traceable to the initial segmentation of the image. We also present alterations to a general segmentation method which reduces the time required for segmentation with this method by about a factor of 10. Several other alterations are given so that this procedure can be used effectively with monochromatic images. We present a method for symbolic change analysis which solves many of the problems encountered by signal based change analysis systems. The problems include the analysis of images with changes in the point of view of the observer, analysis of multi-spectral images without a corresponding increase in the complexity of the analysis, and effective analysis of the detected changes in the image.

Acknowledgements

I wish to first thank my advisor Professor Raj Reddy who has provided inspiration and support throughout the time of the work on this thesis. I would also like to thank the members of my thesis committee, Professors Kendall Preston, Samuel Fuller, and Victor Lesser, for their constructive criticism of this dissertation. Finally I wish to thank the many members of the Department of Computer Science who have provided the environment which was so important for the completion of this work.

TABLE OF CONTENTS

APPENDIX PAGE

The figure numbers below give the chapter number followed by the number that appears under each figure within the chapter.

Figure List

1 The Problem

To date, computer scene analysis has been directed either toward the development of a general system for image understanding comparable to the ability of a human being, or toward the use of a computer to solve a specific well defined problem. This work is intended to be a step toward a general image understanding system rather than a method for the solution of a specific problem. We will describe a system for the analysis of multiple views of a scene to determine what changes have occurred between the views. This work is motivated by the fact that human vision analyzes a dynamic world in order to make changes in the observer's model for what is seen, based on changes in the actual visual world (Gibson, 1950).

This work in change analysis differs from earlier computer change analysis work in the use of symbolic analysis of the image to detect and express the changes which have occurred. Earlier efforts in the change analysis area (Quam, 1971; Lillestrand, 1972; Allen et al., 1973) used correlation guided matching to establish a set of corresponding point pairs. These point pairs are then used to transform the second image so that it is precisely aligned with the first. The aligned images are subtracted and changes are indicated by a large difference in the intensity value of the point in the two images. That is, two images are processed to produce a third image which indicates possible changes. We propose that change results should be presented symbolically. Rather than generate a symbolic description of the difference image, which is not always reliable, we also propose that the initial matching should be done symbolically. The use of symbolic analysis is intended to expand the class of images which can be successfully analyzed for changes compared to the class of images processed by techniques depending on point to point matching and global transformations.

Correlation guided matching has also been applied to many other problems such as stereo analysis (Hanna, 1974; Levine, 1973), and tracking weather echos through many sets of radar data (Duda et al., 1972; Blackmer et al., 1973). The work of Balder(1975) on symbolic motion analysis used completely and correctly segmented images (i.e. done by a human operator, not by machine). We will be using "real" images which will require the generation of the symbolic descriptions in addition to the processing of the symbolic descriptions.

Before the symbolic analysis can proceed, we must first produce the suitable symbolic descriptions of the image. A symbolic description of an image is composed of the regions which make up the image, and the features which describe the regions. There have been several systems designed to segment "natural" images into separate regions. Two of these techniques, region growing (Barrow and Popplestone, 1971; Yakimovsky, 1973) and region splitting (Ohlander, 1975) have been applied to different type of images. The region splitting technique uses less outside knowledge about the content of the scene for the generation of the segmentation. We will use the basic region splitting technique for segmentation, but because this technique is "slow", we will propose several alterations to this method so that it will be more effective. The major alteration is the use of "planning" (Kelly, 1971). The planning method is also based on the structure of the human eye where the peripheral area guides the detailed processing (by the fovea) by larger scale processing. In this same area, Hanson et al.(1974, 1975) are extending the planning concept to all the processing done on the image.

The use of symbolic methods for the analysis of images is not new. Many systems have worked only with the symbolic descriptions (Guzman, 1968; Balder, 1975), but we must derive the symbolic description from the segmentation. Features for describing the regions should be features which are useful both for expressing the change results, and for matching or recognition processing. Since these are features which could also be used in a general image understanding system, we feel that the features should be the same type that are used by humans for the same type of processing (Akin and Reddy, 1976). These feature classes (size, shape, etc.) will be used as a guide to describe the actual features which we will compute. The particular features which we use are derived from many sources, especially from Tenenbaum et al.(1974, 1976) and Duda et al.(1972). There are other methods for the symbolic description of three-dimensional objects and scenes. We do not intend to generate a three-dimensional representation the object which was the goal of the research by Agin(1972) using data generated by a range finder or Baumgard (1974) using controlled multiple views of simple objects.

1.1 Organization

The next chapter will discuss the above research in more detail. Much of the work of these researchers will also be discussed in later chapters, but there the emphasis will be more on their approach, what they accomplished, and how we used their work.

Next we describe the images which will be used for analysis, give the tasks to be performed on each image, and the type of outside knowledge necessary to perform the task. This chapter also discusses the hardware and software used for this work.

The fourth chapter discusses the segmentation of images of natural scenes into their basic regions. Several examples are presented with summaries of the computation times required.

The next chapter discusses the types of features which we use in the analysis, and the methods for the computation of these features.

Next we describe our method for the symbolic analysis of multiple images. Several examples are given which will illustrate the matching process for simple well segmented scenes with a few regions (around fifty), and the matching of more complex images with many, sometimes similiar, regions.

The final chapter gives a summary of the important results and describes directions for further research.

The appendices contain descriptions of some of the programs and operators which are mentioned in the main body of this thesis. The times required for certain operations and other information which may be useful in understanding various operations is also presented. The appendices also contain a detailed description of the processing required for the performance of one of the tasks. We will also present more details of the matching and change results, showing what exact features were used and how these features contributed to the matching operation, and how these features changed.

Figures are numbered sequentially within each chapter. References to a figure within the chapter are by the figure number alone (e.g. Figure 17), and references to figures in other chapters are by both chapter and figure number (e.g. Figure 19.17).

2 Background

This chapter will present a survey of the past work in computer vision which is relevant to this work. This includes work in the segmentation of natural images, symbolic description of images, and change analysis.

2.1 Segmentation

The segmentation of "natural" scenes (e.g. houses, roadsides, people, animals, etc.) presented problems not encountered in the analysis of block-like images which thus required the development of new techniques for image segmentation. Unlike block scenes, "real" world scenes contain many distinct regions, with many different shapes, with few straight edges (except for man-made objects), and with highly textured areas. Two of the new techniques for the segmentation of natural scenes are region growing and region splitting. We will present several segmentation techniques which have been used in the past for a variety of images.

Roberts (1963)

No discussion of past work in computer scene analysis is really complete without a mention of the work of Roberts. Research in the analysis of three-dimensional scenes began with his analysis of block-like objects. Many successful later efforts used methods and systems very similiar to those of his early effort. The Roberts system is an example of a complete computer scene analysis system - it used pictures for input, applied preprocessors to detect the important features, recognized the objects, and manipulated the final recognized objects.

The important feature of block-like objects is the edge (change in intensity) between two faces of one block, two faces of different blocks, or between a block and the background. The edge is important since blocks can be easily and simply represented by line drawings with lines representing edges. The preprocessor, which indicates that an edge may be present, is imperfect (partly because the data itself is imperfect), and extra edges may be located and some edges may be missing. Because of this, the edge data must be processed to collect groups of edges into lines, remove small segments, and extend longer segments until they intersect. This processing will produce a complete (or at least sufficient) line drawing of the scene.

The line drawing is then processed to extract the three-dimensional objects in the scene. The representation of the scene is compared with models of the possible objects (cubes, wedges, and hexagonal prisms). When an object is recognized, it is removed from the representation of the scene so that it will not interfere with further matches. The models can be rotated or scaled in any dimension so that they will match any similiar object. The final three-dimensional representation can be displayed graphically, and individual objects manipulated (moved, removed, etc.) by a graphics system.

Many later researchers have developed one (or more) of these areas - by finding better line drawings, by processing line drawings to recognize objects, or by extending the manipulative capabilities of the computer system (e.g. robotics) - with most of the later efforts in block-like objects patterned after Roberts' work.

Waltz (1972)

Waltz also worked with line drawings of blocks, but these drawings could also include shadows. Waltz classified all types of possible vertices to indicate the possible interpretations (i.e. which faces were in the same or different blocks). The program started by assigning all possible interpretations to one vertex. Then it made an assignment of all possible interpretations for an adjacent vertex and eliminated all inconsistent interpretations (those that were included in one, but impossible for the other). This "filtering" step is continued at all successive adjacent vertices until there is an assignment for all vertices and all inconsistencies have been eliminated. This procedure can possibly yield two or more interpretations, in which case the figure is ambiguous and both are returned.

This program showed that the segmentation of perfect (or nearly perfect) line drawings of block-like objects could be reduced to an algorithmic process. This program is the culmination of the effort in analysis of perfect line drawings of block scenes and leaves very little left undone in this area.

Barrow and Popplestone (1971)

At the University of Edinburgh, Barrow and Popplestone studied nonplanar objects while working on the robot project. The initial analysis produces a set of regions with a small range of brightness values. The final regions are grown from the initial regions by adding small regions to larger regions, and by combining adjacent regions with low contrast at their common border.

The objects are recognized by comparing features of each region with models of the known objects. The models are derived by processing scenes containing the object in the same manner as the processing required for recognition.

The generation of good regions is limited by the quality of the input (shadows, occlusions, etc.) and will work with single objects only.

Yakimovsky (1973)

While studying the general problem of navigation of a vehicle on an outdoor roadway, Yakimovsky developed a system to understand single road scenes. The basic method used was the generation of regions with similiar features and the interpretation of these regions based on a world model.

The regions were grown from original seed regions created by simply dividing the picture into small squares. (Except for time and space limits a one pixel seed region could be used.) Boundaries between regions were eliminated if the "difference" between the two regions at that boundary was below some limit. The difference is calculated both from the difference in image information (such as color and intensity) between the two regions, and from the length of the boundary between the two regions.

The final merging of regions and the assignment of meaning to the regions is based on the probability that a region is part of a particular feature using the information in the world model.

The basic system was sufficient to use on road scenes and, with a different model, on cardiac angiograms, but there is still the need for a training session for each new type of picture to put new probability distributions in the model. The system had no provisions for intergrating a sequence of images of a scene into one representation, but the new probabilities determined by early images in the sequence could be used to aid in the interpretation of the later images. Because of the necessity to divide the picture into basic regions which are larger than one pixel, small thin features may be missed and an edge finding step was needed to obtain an accurate outline of each feature.

Tomita et al. (1973)

Researchers at Osaka University in Japan explored a method of segmenting scenes based on the structural analysis of textures. The scenes studied were artificially constructed by arranging simple black patterns (squares, dots, triangles, etc.) on a white background. The preprocessor extracted all these basic regions which are then used in the further analysis. Larger regions were then extracted by removing groups of similiar basic regions. The properties for segmentation were selected by analysis of the histograms of the features of the basic regions such as size, density, and shapes.

Ohlander (1975)

At Carnegie-Mellon University, Ohlander did some preliminary work on the development of a general image understanding system. One of his main areas of research was one of the major problems in understanding natural scenes: the segmentation of the image into meaningful objects.

About thirty pictures of six different types of scenes (indoor scenes, people, animals, houses, cars, and cityscapes) were photographed and one of each type was selected for experimentation. Each of the scenes was digitized to about one half million points for each of the three colors (red, green, and blue). Initial experimentation showed that techniques which produced results in scenes with blocks break down completely in natural scenes, which contain few straight lines, many heavily textured areas, and indistinct edges.

One feature of natural scenes is that an natural "object" is usually homogeneous in some property such as textural characteristics, color, surface orientation, or depth. Ohlander developed a method to use this property of homogeneity to split the image into separate regions, which could then be associated with objects. By plotting a histogram of the distribution of values for the various features, objects appeared as peaks in the distribution for some feature. The separation of objects by peaks in the histogram is easily seen in simple scenes (e. g. surface orientation of blocks), but in complex scenes the feature values overlap and several objects may have similiar values.

The primary method used by Ohlander was to split the picture (or subpicture) into two parts, one of which represents the peak of some feature, and the other, the remainder of the picture (or subpicture). Then the points corresponding to the peak are further analyzed to determine if this region can be divided in the same way using one of the other available parameters. In many cases, multiple objects that have

similiar properties can be separated by spatial analysis (i.e. they form several distinct regions). The separation continues until there are no features with more than one peak, or until the regions generated are below a threshold. Each of the separated objects (and intermediate segmentations) is represented as a bit mask that indicates which points in the picture are contained in the region.

On many (lightly textured) scenes this method works very well as is, but a textured area will usually exhibit a distribution that indicates a possible region split, but does not yield meaningful connected regions. For example, a bimodal distribution can be caused by the two or more colors (or intensities) that generate the textural elements. To avoid this problem, he introduced a texture measure which indicated those areas which were heavily textured. These areas could be either separated by texture (i.e. one or more textured regions in a relatively homogeneous region), or subdivided by other parameters after further processing such as smoothing to eliminate the effects of texture.

This system was able to obtain good segmentations of several very different natural scenes, but the system as presented required considerable human interaction. Most of the required interaction involved peak finding and selection, the selection of connected regions, and the maintenance of the data base - all of which computers should do well, so that human interaction can be limited to verification and guidance in new (or difficult) situations.

Another source of the cost (time) required for this algorithm was the use of large pictures. Large pictures are necessary for textural information and for accurate location of objects. Ohlander also discussed that if a general description of the large objects is all that is desired then it would be reasonable to use reduced pictures. The reduced regions (a "plan") can then be used to obtain accurate definitions of the object in the original picture.

Lastly, Ohlander also described problems concerning shadows (or highlights) and occlusions, principally the problems of detection and removal of these distortions.

Kelly (1971)

Kelly developed a system for distinguishing pictures of people, using a picture of both the face and the entire body. The system worked by first finding the most obvious feature - the body or face outline. This feature location is then used to locate the next most obvious feature, and so on, until all desired features are located. The identification is then based on the feature locations (or rather the distances between feature locations, or the size of the features).

The location to the individual features is done by special heuristics using knowledge of the appearance of the feature. For example, the mouth is located with a simple line-finding algorithm which locates the dark line between the lips, and the eyes are located by the intensity of a scan line across the eye (the dark iris is surrounded by the lighter white of the eye-ball and the white of the eye is then bordered by the darker skin of the face). The program depended heavily on the location of the early features for the application of heuristics for the later features.

An important aspect of the feature location was the use of "planning". To

locate the outline of the first feature, the head, a reduced picture is used to obtain an approximate location. The reduced picture allows the program to do searching and backup without incurring a heavy time penalty. The reduced picture also smooths out small defects in the input picture due to noise, lighting, or background objects.

This program depended on a reasonably clear picture of a person which conformed to the expected model (i.e. no glasses, hair not too long, no beards for some features). Because of the use of special heuristics, it would be difficult to extend the program to handle pictures which do not conform to the current model.

Hanson et al. (1974, 1975)

There is a current effort at the University of Massachusetts to develop a system to analyze natural scenes. The principal paradigm is to apply an operator on one image to reduce its size (for example, by half in each dimension) and to use results derived on the smaller images as guides for locating the features in the larger images. Other functions, or even the same ones, can be applied at a single level, or on several images at one level (this is called an iteration stem). Possible operators include a gradient operator, average of a window, maximum in a window, minimum in window, normalize three-color image, generate color features (hue, saturation, and intensity), etc.

The application of several of these operators is used to locate relevant features in the image, such as lines, edges, spectral feature values, textures, and regions. Regions grown in the smaller pictures are then projected back to the larger images.

Models from Human Vision

The human eye has two types of receptors, rods and cones. The rods are used for black and white vision and react when one of their rhodopsin molecules is hit by a few photons. Rods are distributed very unevenly on the retina, with the greatest density near the fovea (there are no rods in the fovea), and rapidly decreasing densities away from the fovea. Cones are used for color vision and are concentrated in the fovea. There are three types of cones with red, blue, and yellow-green sensitivity peaks.

Visual acuity is best in the foveal region, caused not only by the increased density of receptors, but also by the increased number of nerve cells (bipolar and ganglion) per receptor in this region. Since acuity is relatively poor in the periphery, the eye must be moved so that areas of interest are projected on the foveal area for detailed analysis. The peripheral area is sensitive to motion and changes, and directs the eye to study areas with many edges.

2.1.1 Segmentation Summary

We will be using the region splitting techniques described by Ohlander. This method was originally developed for color images and is very slow. We will modify the segmentation procedure to operate on monochromatic images by the use of simple textural measures. Planning techniques will be used to make the segmentation computationally more efficient.

2.2 Features and Symbolic Descriptions

In the past, symbolic analysis has been extensively applied to simple block-like objects and, to a lesser extent, with natural images. The symbolic representation can either be a feature based description as used in human vision, or representational as a set of simple three-dimensional objects as is used with blocks.

Akin and Reddy (1976)

At Carnegie-Mellon University there has been some research into what features are used by people when analyzing scenes (images). These experiments used recorded protocols of human subjects analyzing an image. The subjects worked under many of the same constraints that a computer must work under. The subjects did not view the image directly, but were allowed to ask the experimenter questions about the image, which the experimenter answered by looking at the image. In each of the experiments there was a particular task which provided some guidance to the subject. Some of the tasks were to describe the scene; to select the picture from a set of twenty pictures; and using a map and questions, to find a location on the photograph which the experimenter selected.

The experiments showed that people commonly use a limited number of feature extraction primitives in classes such as size, shape, location, quantity, color, and texture to analyze the scene.

Tenenbaum et al. (1974, 1976)

At the Stanford Research Institute there has been research on the design of an interactive system to be used for research in scene analysis. One class of scenes that has been used in this work is office scenes. The system allowed the user to interactively select portions of the scene which have certain features, and to generate descriptions of the object from these features. Possible features are color and intensity information, height, depth, and surface orientation. The system is not designed to identify all objects, but merely to locate specified objects.

The description of the features of the object includes information about which features are the most important for the location of the object. By using an easily found feature first, the potential search space can be greatly reduced, and the object might even be located by this simple feature. After this initial feature location, the selected object or objects are then verified using the more expensive features.

In addition there has been other research at SRI on a procedure for the interpretation of scenes using a "filtering" technique similiar to what Waltz used on blocks. The filtering process is combined with a region growing procedure to generate a segmentation and interpretation of each scene. Initial regions are generated by grouping all identical adjacent points into an initial region, these regions are assigned possible interpretations (e.g. all interpretations). The filtering step is applied to eliminate inconsistent interpretations. After each application the the filtering procedure, all adjacent regions with identical interpretations are merged together, then the filtering is applied again.

An interactive approach such as this easily allows the development of models of possible objects, the testing of ideas of how to recognize objects, and the exploration of the recognition of objects in a limited environment. This type of effort can lead to a better understanding of what is required for the analysis of natural environments.

Agin (1972)

Agin worked on generating three-dimensional representations of simple objects (a doll, glove, toy horse) using a laser ranging system coupled with a TV camera for input. The resulting representation consisted of circular cross sections about several axes.

Baumgard (1974)

Baumgard studied the generation of three dimensional representations of simple objects by using the intersection of several conical representations of the objects. Each conical representation is the locus of all possible objects which could generate one of the two-dimensional images. The set of images was obtained by placing the object on a turntable and taking several lateral images at different rotations.

Several other areas of research had to be explored before this work could be done, including the generation of object outlines, the matching of features of the outlines, and the generation and manipulation (computation of intersections etc.) of polygonal representations of solids.

2.2.1 Feature Summary

We will use features of the same type that are used in human analysis of two dimensional scenes (Akin and Reddy) rather than the three-dimensional representations of Agin and Baumgard. The actual measures which we will use to represent features in these classes are derived from many different sources. Some of the feature measures are obvious, such as the size of the segment. Many of the measures were taken from Tenenbaum et al. where they were used as descriptors for individual textural elements. Other measures were taken from Duda et al.(1972). This last reference is discussed in the next section on matching.

2.3 Matching and Change Analysis

All the past change analysis systems which use image data have used signal based matching techniques, and have produced an image as the change result. Symbolic change analysis has been restricted to the analysis of scenes which are already segmented and described (i.e. correctly represented symbolically).

Levine et al. (1973)

Another project in computer vision prompted by the needs of space exploration is the Mars rover project at the Jet Propulsion Laboratory. This vehicle

must operate for extended periods without human guidance and must be able to travel between two points autonomously. One of the important features for navigation in the Martian environment is the distance from the rover to points on the surface in front of it. This distance (range) information can be used to detect cliffs (extreme distances) which must be avoided, rocks (large and small) which may interfere with travel, or relatively smooth areas which are good for travel.

JPL's method uses the parallax shift determined from images from fixed stereo cameras to derive a depth map for the scene in front of the vehicle. Since fixed stereo cameras are used, the search for corresponding points can be reduced to a search along one scan line in the television image of the second view. To eliminate the fruitless matching in large homogeneous regions, only points in the first image that are along edges (e.g. between a rock and the background) are considered for matching.

This research has been directed more toward a reliable solution to the navigation problem than toward basic research in image understanding, but the analysis provided by the stereo camera system can be used in a more complete image understanding system.

Hanna (1974)

After the results of Quam and others showed that computer matching of pictures was possible and useful, it became important to consider more efficient methods to derive this match. Hanna discussed several different matching functions (correlation, RMS error, etc.) but it is apparent that the best way to improve efficiency is to reduce the number of matching operations that are required.

Hanna explored several methods for this. One is to use a fast pretest for a likely match in a neighborhood (e.g. by comparing the variance or average values). This will eliminate obvious mismatches and can also be used to sort areas by the likelihood of containing the best match.

She also discussed growing regions of constant (or near constant) parallax by testing points adjacent to known points for the same parallax shift. These regions of constant parallax can be used to hypothesize surfaces and objects. The camera location can also be used to restrict the search to a single line through the image (as was used by Levine et al. (1973)).

Since the camera locations may not be known, she also explored the derivation of a camera model from a set of corresponding points. The program iterated on the camera model trying to reduce the error between the expected and actual point locations. The derivation of the camera model is not as reliable and accurate as would be desired, however, for depth calculations.

One area left to future researchers was that of matching regions in the picture rather than of single points.

Duda et al. (1972, 1973)

A group at the Stanford Research Institute has applied pattern recognition techniques to the problem of tracking storm cells in digitized weather radar data.

Some of the problems studied were the consistent extraction of individual cells in a line of storm cells, matching cells in consecutive images, and forecasting the position of the cell in the next image.

The cells are located by applying a high threshold to the image (the echo intensities range from 0 to 9), and then extending these cells to include adjacent points with a value one less than the initial threshold. These initial regions are merged into a single cell only if the extension step causes them to add the same point; therefore two cells may be adjacent. This procedure proved more reliable than simple thresholding for this task.

The matching procedure proceeds in two steps. First all echos are translated by their expected motion and then a global correction is determined by searching for the best correction, using a simple cross-correlation method. Then each echo is translated for a best match within a neighborhood of the location given by the global correction. This limited search is used to prevent two echos from matching the same echo in the new image.

The prediction program uses the past velocities to approximate the new locations. Because of fluctuations in the velocity values and the unresponsiveness of arithmetic smoothing to sudden changes, they used an exponentially weighted averaging method. New echos receive an initial velocity based on nearby echos with more weight given to older echos.

The earlier report also discusses echo description, particularly the description of the contour. The contour is represented as two periodic functions, one for the X coordinates and one for the Y coordinates. The functions are then represented by Fourier approximations, which take less space to store than the entire contour.

Quam (1971)

With the space program came a need for the analysis of many pictures. To aid this, Lynn Quam worked on a system to compare two images taken at different times or different locations by the Mariner spacecraft in orbit around Mars. This comparison causes some features to become more apparent that they were in a single image. Features such as cliffs, canyons, etc. may not be readily apparent in a single image, but could have a very different appearance in two different images. Because of the conditions on Mars at the time the pictures were taken, there were also changes due to dust, from a dust storm, settling around various features.

As a first step, a set of corresponding points in the two images is located. The program used points on a grid in one picture and found the best match in the other picture using the correlation coefficient of the neighborhoods of the points. The two images were known, *a priori*, to be of the same general area and initial transformations were applied to one image by using the known satellite locations and camera transformations, but the orbit locations were not known well enough to use this transformation as the final result. Based on the discrepancies determined in the match, a final transformation (rotation, translation, etc.) is calculated to minimize the error between the two pictures.

By making a difference picture (between the initial image and the transformed

second image), the areas that are different in the two views can be located. Most of the two images will be approximately the same (a difference close to zero), but some features may cause areas of change and these areas will have large difference values.

This system was intended to aid a human in studying the pictures from Mars, so there was no need for completely autonomous image analysis and no need for real time results.

Lillestrand (1972) and Allen et al. (1973)

At Control Data Corporation, there has been work in developing a computer system for the detection of changes between two images of the same area. The basic system is a collection of special processors connected so that the two images are processed in a pipeline. Each stage of the pipe does one major operation, such as the search for corresponding points, transforming the images, subtraction, etc.

The differences between this system and Quam's is primarily in the method of transforming the second image. The CDC system transforms smaller portions of the image separately. A quadrilateral in the second image that corresponds to a square in the first image is located (by finding the corresponding points for the four corner points). This means that the picture can be processed sequentially in one pass through the pipeline.

The base image and the transformed image are subtracted and differences are analyzed by a human operator. Some differences can be automatically analyzed and noted as being uninteresting because they are shadows or highlights.

This method depends on a spectral match of the two images and on a global transformation of the data. Because of relative position changes inherent in near and medium field multiple images, a global transformation of a picture or a portion of the picture to align it with another image would not produce meaningful results.

Balder (1975)

Balder developed a system to produce a linguistic description of the motion in a sequence of images. The input is a sequence of images, which are already segmented into primitive regions and objects. This initial data base also contains feature locations and relations which might be derived from a single image. From this sequence the system produces a correct English language conceptual <u>description</u> of motions in terms of trajectories (translations) and rotations of the objects or the observer. The resulting motion descriptions and relations in the data base are simple, but sufficient to describe the sequence. The motion of objects is restricted only by the fact that the objects are natural, the scenes were taken on the earth (so gravity affects motions), and that the observer is passive and human-like.

A description of the motion differs from an explanation of the motion or an understanding of the sequence in that an understanding requires high level knowledge about the type of scene, the environment, and the intended use for the explanation of the sequence.

Balder made several assumptions about the type of scenes that the system

would handle. The sequence consists of discrete, static images. This is a practical restriction; it is difficult to obtain, store, and process continuous pictorial information. The sampling rate of the discrete images restricts the maximum frequencies of oscillations that can be correctly detected and analyzed. The sequence contains only recognizable objects in natural environments (i.e. there are no tigers in offices; no optical illusions). The allowable motions are only rigid motions (rotations and translations of the object or a subpart of an object), but the observer is also allowed to move. If observer motion is not known, it can be deduced from the movement of fixed objects. Likewise, the fact that an object can move may be contained in the description of the object, but it may also be derived from the movement of the object.

The representation of motion uses the same type of structure as the representation of objects. Models of all objects are represented as a graph structure with the nodes representing parts of the object (or entire objects) and edges of the graph representing relations between objects; if an edge begins and ends at the same node, it represents a property of the object. Object properties and relations include the type, the subparts, the location, the orientation, and the size of the object.

Motion in the scene is represented as "events". Each event represents one sequence of continuous motions (or repetitive motions). Event properties include the subject and agent of the motion event (which object moved or which was moved by another object), the direction, trajectory and axis of the motion, and possibly an indication of the next event in the sequence that is needed to describe the motion.

This initial event structure is much too long and repetitious for the purposes of a simple linguistic description of the changes. This description is condensed by the use of "demons" which are activated by the presence of certain preconditions and transform the representation in various ways. By the use of these demons the event descriptions are simplified by changing the long descriptions into more natural English-like sentences by modifying the event descriptions into verbs and adverbials which are commonly used to describe motions and directions in English.

This system was able to describe the motion in several sequences in correct, relatively concise English, but because of the first assumption (that the scene was already segmented and recognized), it has little immediate application to the analysis of "live" dynamic natural scenes. It showed that motion can be detected and described in an already well-understood set of images (as Guzman(1968) and Waltz have done with blocks), but does not address the problems of using the motion and change information to reduce the processing necessary to understand a sequence of images of one scene.

2.3.1 *Matching Summary*

The correlation based matching and change analysis systems perform well on a limited set of images. But, when the images are taken from very views, the correlation matching is unreliable, and when there are changes in the number of objects in the scene (new or missing objects) the matching is impossible. Many changes that occur in a scene require higher level processing to analyze. Because of these problems we will attack the matching and change analysis problem at a symbolic level rather than at the image level. Balder has shown that understanding (in a limited sense) is possible with completely and correctly segmented scenes, but these results will not necessarily apply when there are errors in the machine generated segmentation.

3 Data and System Description

This chapter describes the images which are used for analysis, the tasks which are to be applied to the sets of images, and a description of the outside knowledge necessary for the tasks. The chapter also describes some of the hardware and software support for the work.

3.1 Images

3.1.1 Representation

We represent images as a matrix with an arbitrary number of rows and columns where each picture element (called a pixel) can be from zero to thirty-six bits long (limited by the machine word size) and pixels are packed as many as possible into a word. Each image also requires an indication of the relative offset from the original image, if it is really a subimage. The top left point in the image is pixel[1,1] and the bottom right point is pixel[number of rows, number of columns]. Picture points are referenced to by "I" and "J" coordinates, i.e. pixel[I,J].

Since images are an arbitrary size, it is not usually possible to hold the entire image in primary memory. Thus we have implemented a system where portions of the picture (individual rows) are read from secondary memory when needed. The system automatically decides if it is necessary to page the picture or if the picture can be maintained in primary memory. A small number of recently accessed rows are maintained in core and are written back on disk (when removed from core) only if changes have been made.

3.1.2 Scenes to Analyze

A short description of all the images used is given in Figure 1. This figure shows the amount of data for each scene in terms of the number of rows, columns, the number of bits per pixel, the number of spectral bands per image, and the number of images per scene. The types of camera induced changes are also given.

Scene Name	Rows	Cols	Bit	Band	Images	Distance	Camera Motion	Figures
House	725	748	8	3	2	12 M	3 Meters to the left	2,3
Cityscape	725	748	8	3	2	1 Km	50 Meters to the left	4,5
LANDSAT	2400	3200	6,7	4	2	900 Km	18 Days in orbit	6,7
Rural	2000	1900	6	1	3	--	Rotation	8,9,10
SLR	2000	1800	6	1	2	--	Translation	11,12
Urban	2000	2000	8	1	2	--	Translation and distance	13,14

Figure 1 Image Descriptions

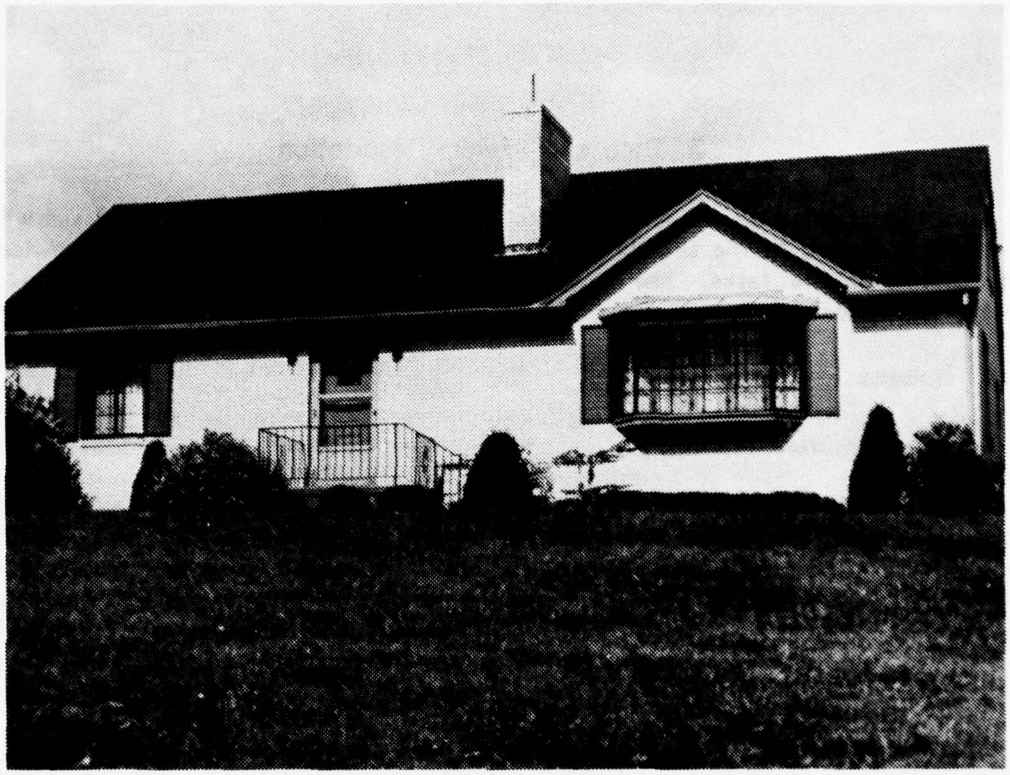


Figure 2 House 1

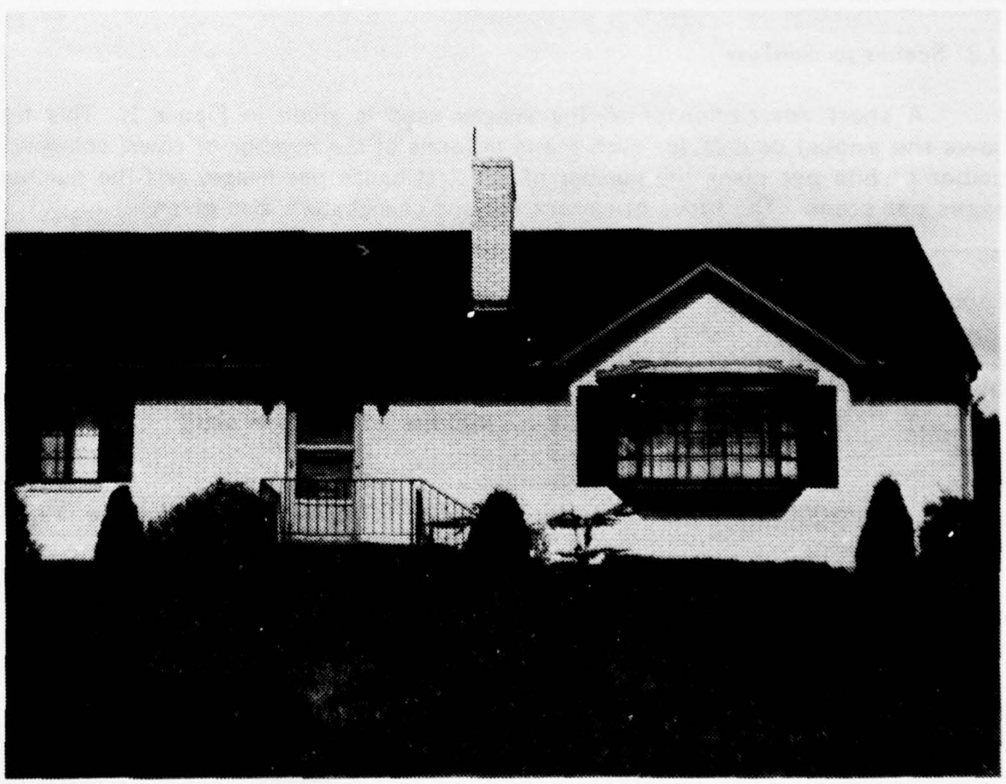


Figure 3 House 2

Figure 4 Cityscape 1

Figure 5 Cityscape 2

21 MAY 1973

Figure 6 LANDSAT 1

8 JUNE 1973

Figure 7 LANDSAT 2

Figure 8 Rural 1

Figure 9 Rural 2

Figure 10 Rural 3

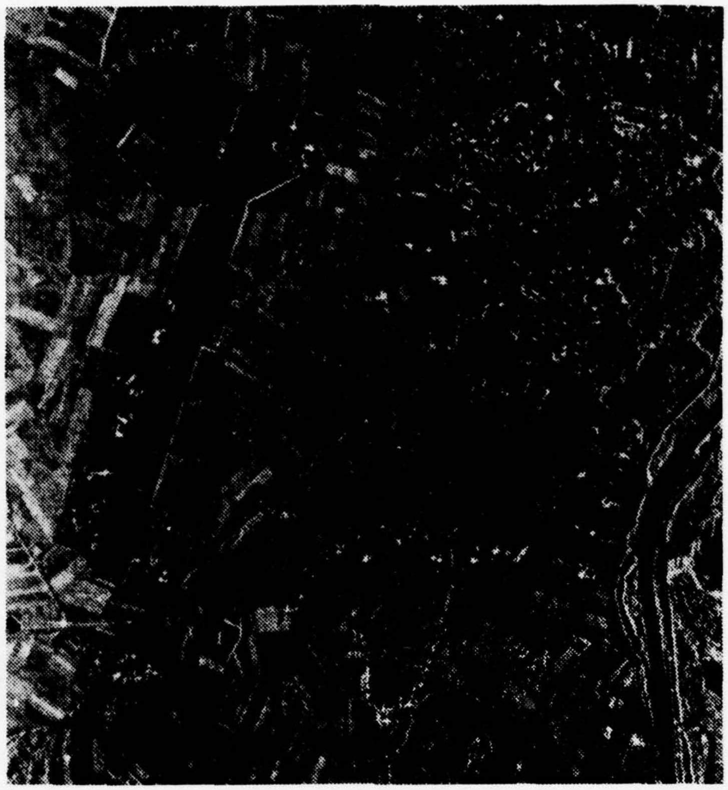


Figure 11 SLR 1

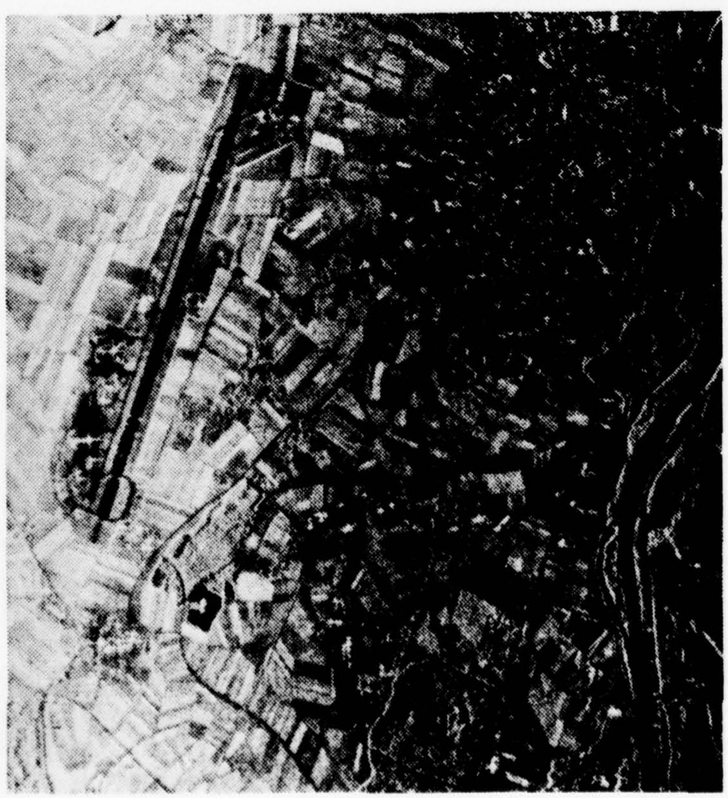


Figure 12 SLR 2

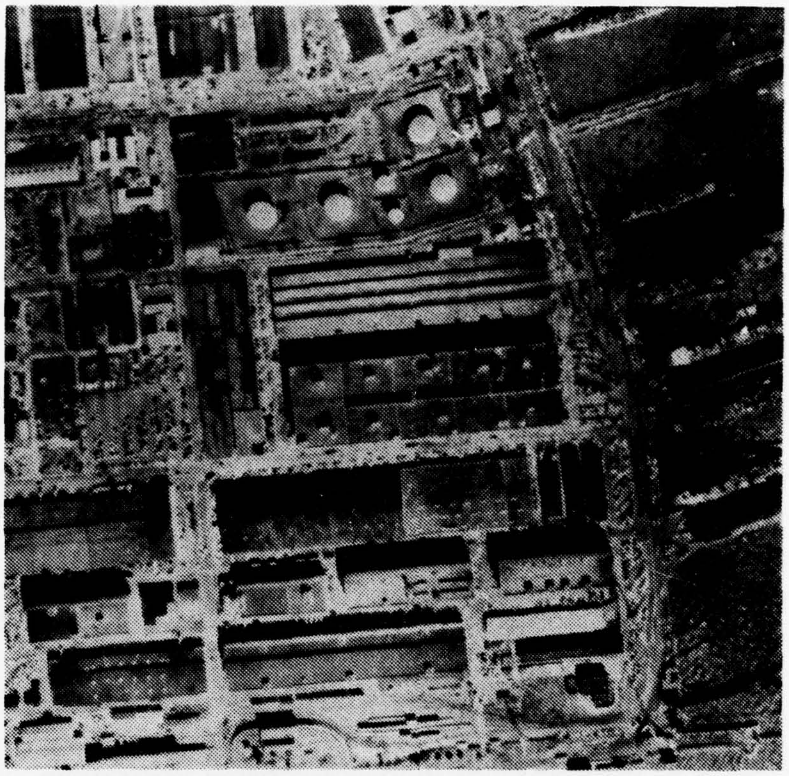


Figure 13 Urban 1

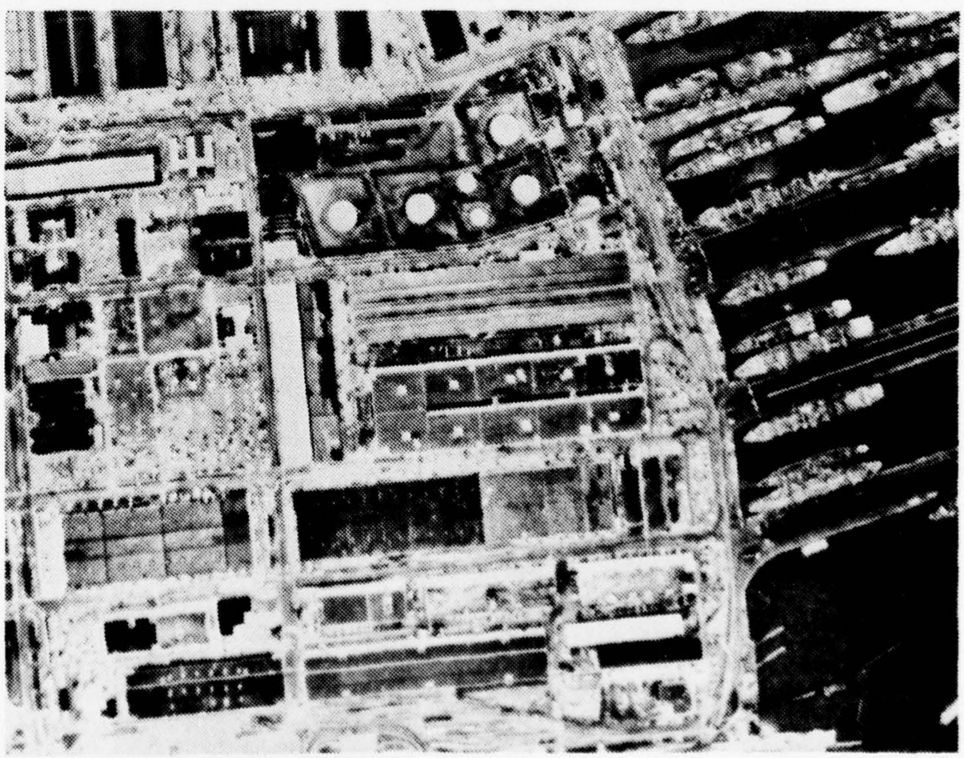


Figure 14 Urban 2

The House and Cityscape scenes are digitized from color prints[1]. These pictures were made specifically for the purpose of studying changes between images which are introduced by a change in the camera location. The house was selected because is is not surrounded by trees and is clearly visible. The cityscape is of a portion of downtown Pittsburgh. These scenes contain large (relative to the image size) and generally well defined regions with varying amounts of textural information - the cityscape scene has much more textural variation than the house scene. The three spectral bands for these images are the red, green, and blue intensities in the color image. The digitized images do not include the entire photograph as shown in the figures; in all four images the left edge of the image is cut off (the house image ends at the left window and the cityscape just beyond the left edge of the large building in the left center of the picture).

The LANDSAT scene is of the Wind River, Wyoming area[2]. These images were generated by the multi-spectral scanner (MSS) of LANDSAT 1. This satellite completes its coverage of the earth every eighteen days, so that these images are of the same area but there are some differences in the area covered since the satellite position is not that precisely controlled. Each pixel in the image corresponds to a 50 meter by 80 meter area (about one acre) on the surface. The four spectral bands of these images correspond to green, red, and two infra-red ranges. The two images are printed (see Figures 6 and 7) so that they line up with the surface, but are stored as rectangular arrays.

The rural scene is represented by three monochromatic aerial photographs[3]. These images contain several large, smooth (untextured) regions, and many more small bright regions. Unlike our other images, bright points in these images have values near zero rather than near the maximum value. The first few columns on the left side contain dark points which will introduce spurious information when histograms of the entire image are generated.

The SLR (side looking radar) scene introduces a completely different spectral domain. A SLR image is bright where the surface reflects the radar signal back toward the source, so the image will tend to get darker further away from the source (i.e. from the left to the right in the image). For example, a smooth water surface reflects the radar signal very well so that it will be bright when directly under the source and dark away from the source. Most of the points in these images (especially the first one - Figure 11) fall within a four bit range rather than the entire six bit range, so that the processing is more sensitive to the noise in the image.

The final scene is the urban-industrial scene. These images have many more distinct objects than the others. In addition to the translation differences encountered in other images, there is also a scale difference between these two images.

[1] These images were digitized by the Image Processing Institute at the University of Southern California.

[2] These images were provided by Albert Rango at the Goddard Space Flight Center.

[3] These images, and those of the next two scenes were provided by the Digital Images Systems Division of the Control Data Corporation.

3.1.3 Tasks for Scenes

Image	Task
House	Segment the large clear regions in the two images. Illustrate symbolic matching by finding the corresponding regions in the two images.
Cityscape	Segment the large regions. Illustrate symbolic matching in images where the segmentation has more differences than the House scene.
LANDSAT	Segment and match certain constant features (the lakes) in the two images. Find snow cover changes in one area.
Rural	Apply the matching process with images that are rotated with respect to each other. The three images allow matching at an intermediate rotation and a more extreme rotation.
SLR	Segment and match several regions in a different spectral domain.
Urban	First: Segment and match certain anchor features. Second: Analyze the changes (missing or new objects) in a given area of the image.

Figure 15 Tasks.

The tasks to be used for the analysis of the given scenes are outlined in Figure 15. A task description is necessary to determine what processing must be done and to allow some evaluation of the results. All the processing of the images representing the scene is done within the framework of the performance of the task. The imposition of a task on the processing is not new; usually computer image analysis systems are designed for one particular task and are unable to perform any other. The task description will control the type of regions which are segmented, the type of regions that are matched, and what change information is desired.

The house and cityscape scenes have few changes between the images so that the primary task is to illustrate the symbolic matching procedure with a simple scene (the house) and a more complex scene (the cityscape).

The LANDSAT task is a simple example of symbolic matching for use in the registration of two images. The differences in the location of the lakes in the two images can be used for transforming one image to correspond to the other. Once several regions are matched, their locations can be used to guide the matching of the larger snow regions.

The rural scene will be used to show symbolic matching in the presence of rotations. This scene has three images, so that matching and change analysis can be performed on a pair with a small rotation difference, and on a pair with a larger rotation difference. This scene will also be used to introduce many of the problems of processing large, monochromatic, aerial images (and how they are solved).

The SLR images will be used as an example of segmentation and matching in a very different spectral domain.

The urban-industrial images have the most complex tasks: the detection of new or missing objects in a given area of the scene. Since this requires limiting the area of the two images being analyzed, and determining the size and position differences, the first task is the location and matching of several specific anchor regions in the two images. Since the final task is to determine the number (change in number) of ships in the pier area on the right hand side of the image, we will need to determine whether a region is a ship, water, or pier region rather than whether the region matches another unidentified region.

3.1.4 Knowledge for the Tasks

For any computer solution of any significant problem in image understanding, some outside knowledge is necessary to guide the processing. This knowledge is implicit in the statement of the task and description of the data, or is implicitely required for the completion of the task. This subsection will describe the knowledge which has been assumed by the task description, or is required extra knowledge not given in the task statement. The external knowledge necessary for performance of these tasks can be loosely divided into knowledge for segmentation and knowledge for matching or change analysis (see Figures 16 and 17). The segmentation knowledge indicates what type of regions are necessary for the execution of the task, and how they may be derived. The matching knowledge indicates which features in the scene are expected to change, and which are expected to remain constant.

3.1.4.1 Segmentation Knowledge

Scene	Segmentation Knowledge
House	Large regions, need a complete segmentation with a general segmentation method
Cityscape	Large regions, need a complete segmentation with a general segmentation method
LANDSAT	Lakes: low intensity and small regions (1000 out of 6 million points), snow: high intensity and large regions
Rural	Large smooth areas (no edges), bright regions are very small (250 points out of 4 million), some dark regions correspond to image flaws
SLR	Smooth regions, general left to right intensity gradient, textures offer the best chance for segmentation
Urban	Small bright regions for the anchor regions, ships are regions with many edges, piers are dark and untextured, water is untextured, general model of pier area

Figure 16 Segmentation Knowledge

In general, the segmentation knowledge is simply an indication of the type of regions desired, such as the "large regions" for the house and cityscape scenes, the small regions (1000 points out of 6 million) of the LANDSAT scene, and the bright and/or smooth regions of the three monochromatic scenes. This type of knowledge is

used to control the segmentation procedure by limiting the type of regions selected (bright, smooth) or by setting the minimum size of acceptable regions. This type of knowledge can be represented as procedures acting as knowledge sources which can force the segmentation procedure to extract the proper regions, or as parameters to other general programs (such as the size of acceptable regions).

In the LANDSAT images, the lakes will appear as dark regions in the fourth band (infra-red) since the water absorbs the infra-red frequencies. The snow surface reflects all frequencies so that these regions will be bright in all bands. The urban-industrial task will also require scene dependent knowledge about the appearance of the water, pier, and ship regions so that they may be easily segmented. This scene will also use procedural descriptions of the pier area to limit the area of the image analyzed for the change processing.

3.1.4.2 Matching and Change Knowledge

Scene	Matching and Change Analysis Knowledge
House	Few changes
Cityscape	Changes in the relative "J" position of regions
LANDSAT	Small translational changes for the lakes, snow areas change size and shape
Rural	Rotation difference, with minimal location difference at the center
SLR	Translation changes, image intensity differences
Urban	Scale, location, absolute brightness differences, different ships and different number of ships

Figure 17 Matching and Change Knowledge

The matching and change analysis knowledge is used to control which features are to be used for matching and what types of feature changes are desired or likely. This knowledge can be represented with lists of features showing which features can or cannot change. For example, in the house scene there are few changes expected so that all features can be used in matching. In the cityscape scene there are some changes in the "J" position (left to right) of the objects so that all but the features dependent on the "J" position can be used. The LANDSAT images have small translation changes so that the absolute location features can not be used, but the relative positions of the objects remains constant. Also the shape and size of the snow regions changes between the two images, so these features will not be useful to find the match. The rural images are rotated with respect to each other so that orientation and location are likely to change and are not very useful for matching. In the urban scene there are scale, location, and absolute intensity changes so that these features can not be used for matching. But the scale and location differences are the same over the entire image so that these differences (once they are computed) can be used to adjust the feature values for further matches. In the final urban change analysis task there will be changes in the number of objects (some appear and some disappear) so that there will be regions in one image without a corresponding region in

the other image. These changes will also cause the size and shape of the background regions (primarily the water) to change size and shape.

3.2 Computer System

Most of the procedures discussed in the next chapters have been implemented in SAIL (VanLehn, 1973) on a PDP-10 with 256K words of primary memory. There has been no effort to maximize efficiency by resorting to machine coding of the inner loops, but there has been some effort to implement relatively efficient algorithms in SAIL.

All but a few of the preprocessing routines have been incorporated into one interactive program to aid in combining various operations into useful sequences. This program has facilities for running in an automatic mode or a manual mode for testing new operations. Timing information, giving the runtime for each routine (or part of a routine) is collected for each run of the program. The PDP-10 (a KA-10 processor) performs about 0.3 million operations per second and all timing information in the later sections will be presented in terms of the number of operations. These operations counts will be derived from actual timing files, and are not necessarily the ideal numbers given in Appendix 3, since the number of operations reflects one implementation on a particular machine. Some individual operations may require ten or more PDP-10 instructions. Special purpose image processing machines are capable of the equivalent of many millions of PDP-10 type operations per second, but only for a restricted set of operations. Since this is a research effort, we cannot commit major portions of the computation to special purpose processors, but these processors are necessary for the implementation of a practical (i.e. commercial) system.

3.3 Data Storage

The information which is used in the matching process and generated in the segmentation operation must be stored in core when being used, and on secondary storage between runs. We have implemented a set of programs which allow the data base in memory to be dumped onto secondary storage (in a text file) and read from this file back into memory. Figure 18 is an example of the disk file version of the data structure.

While in core, the information structure is stored using the SAIL LEAP facilities, which provides the mechanisms for the manipulation of sets, lists, and "relational triples". The triples are defined as an expression: property⊗region≡value which is read as: the property of the regions has some value. A list is an ordered set so that entries can be referenced by the position in the list. Each image is stored as a list with each region being one entry in the list, and relations between regions (or features of regions) stored with the relational triples. The values of the properties can have many different types, such as strings, arrays, integers, real numbers, or other regions. There is a set of properties provided by the system, but these can be increased by the user.

```
[0]LS1
NUMBER OF REGIONS=15
VERSION NUMBER=8
ROWS BY COLUMNS= 268 BY 403
PROPERTIES=
ILOCV 2
JLOCV 2
TABOVE 8
TBELOW 8
TOLEFT 8
TORIGHT 8
TCMASS 2
TSHAPES 9
REGINNEX 2
REGSECNEX 2
TCOUNT 2
TBORLEN 2
TORIENT 4
TSHHTWR 4
TERMINATION
DESCENDANTS= 1
[1]LS1
0
MASK= LS
ANCESTORS= 0
DESCENDANTS= 2 3 4 5 6 7 8
[2] Region is E21n1
0
MASK= E21n1
ANCESTORS= 1
MDERIVE= PARM: 3 UPTHR: 6 LWTHR: 0
ILOCV=393 , STDEV=686
JLOCV=194 , STDEV=406
TABOVE= 13
TBELOW= 7
TOLEFT= 12
TORIGHT= 10 13
TCMASS=5636 , STDEV=3150
TSHAPES= 3 4
167.1820000 .0000000 117.7883300 .0000000
113.5941600 2.9027340 92.4234940 -2.4965636
29.1531180 -3.0756336 16.4944160 1.2782600
REGINNEX=-58 , STDEV=2
REGSECNEX=-892 , STDEV=7
TCOUNT=322 , STDEV=-126976
TBORLEN=1011 , STDEV=0
TORIENT= .6271295
TSHHTWR= .4578873
[3] Region is E21n2
0
MASK= E21n2
ANCESTORS= 1
MDERIVE= PARM: 3 UPTHR: 6 LWTHR: 0
ILOCV=1497 , STDEV=1656
JLOCV=1 , STDEV=183
TABOVE= 12
TORIGHT= 4 5 9 14
TCMASS=15912 , STDEV=898
TSHAPES= 3 4
89.4555160 .0000000 85.5088970 .0000000
62.2303900 2.8339592 84.6855200 -2.9673612
9.0741684 -2.5833775 1.3264529 -.0496285
REGINNEX=-177 , STDEV=3
REGSECNEX=-1739 , STDEV=7
TCOUNT=176 , STDEV=-32768
TBORLEN=567 , STDEV=0
TORIENT= .9539245
TSHHTWR= .2333418
[4] Region is E21n3
0
MASK= E21n3
ANCESTORS= 1
MDERIVE= PARM: 3 UPTHR: 6 LWTHR: 0
ILOCV=1561 , STDEV=1730
JLOCV=1958 , STDEV=2007
TABOVE= 10
TOLEFT= 3 5 14
TCMASS=16540 , STDEV=19817
TSHAPES= 3 4
86.9710570 .0000000 23.1157890 .0000000
72.0777950 -3.0990821 15.4741540 -1.4406806
2.7665110 1.2194443 8.8764467 2.8976434
REGINNEX=-14 , STDEV=4
REGSECNEX=-645 , STDEV=5
TCOUNT=66 , STDEV=-57344
TBORLEN=380 , STDEV=0
TORIENT=-.0196810
TSHHTWR= .2137845
[5] Region is E21n4
0
MASK= E21n4
ANCESTORS= 1
MDERIVE= PARM: 3 UPTHR: 6 LWTHR: 0
ILOCV=1561 , STDEV=1623
JLOCV=1817 , STDEV=1906
TABOVE= 10
TOLEFT= 3 14
TORIGHT= 4
TCMASS=16003 , STDEV=18635
TSHAPES= 3 4
37.6384980 .0000000 47.6338030 .0000000
24.7168110 -3.0962964 39.7276810 -.3846805
6.9774177 2.4229651 2.1713433 2.4536156
REGINNEX=-12 , STDEV=5
REGSECNEX=-576 , STDEV=8
TCOUNT=24 , STDEV=28672
TBORLEN=213 , STDEV=0
TORIENT=-1.0337020
TSHHTWR= .1940102
[6] Region is E21n5
0
MASK= E21n5
ANCESTORS= 1
MDERIVE= PARM: 3 UPTHR: 6 LWTHR: 0
ILOCV=1150 , STDEV=1205
JLOCV=2777 , STDEV=2848
TOLEFT= 11 12
TCMASS=11785 , STDEV=28112
TSHAPES= 3 4
27.3548390 .0000000 36.6881720 .0000000
23.7526290 2.9791054 30.3854070 -.8036372
2.4661455 -1.4172855 2.1575014 1.0758289
REGINNEX=-1493 , STDEV=5
REGSECNEX=-1670 , STDEV=8
TCOUNT=24 , STDEV=-106496
TBORLEN=186 , STDEV=0
TORIENT=-.9358927
TSHHTWR= .3199369
[7] Region is E21n6
0
MASK= E21n6
ANCESTORS= 1
MDERIVE= PARM: 3 UPTHR: 6 LWTHR: 0
ILOCV=737 , STDEV=777
JLOCV=282 , STDEV=319
TABOVE= 2 13
TOLEFT= 12
TORIGHT= 10
TCMASS=7576 , STDEV=3026
TSHAPES= 3 4
20.5100000 .0000000 20.7800000 .0000000
18.4846930 2.9725267 17.5832330 -1.8485558
.4055067 -.4019679 .4091375 -.4459152
REGINNEX=-55 , STDEV=6
REGSECNEX=-536 , STDEV=7
TCOUNT=16 , STDEV=-126976
TBORLEN=100 , STDEV=0
TORIENT= .5693822
TSHHTWR= .8870293
[8] Region is E21n7
0
MASK= LS
ANCESTORS= 1
DESCENDANTS= 9 10 11 12 13 14 15
[9] Region is e21r10
0
MASK= e21r10
ANCESTORS= 8
MDERIVE= PARM: 3 UPTHR: 64 LWTHR: 34
ILOCV=882 , STDEV=1949
JLOCV=1578 , STDEV=3224
TABOVE= 10
TOLEFT= 3 11 12 14
TCMASS=14554 , STDEV=23561
TSHAPES= 3 4
523.8100900 .0000000 758.1323800 .0000000
402.2344600 2.7531727 655.4478600 -2.8972436
62.2357200 -1.8418536 171.6351300 -.7159144
REGINNEX=-480 , STDEV=10
REGSECNEX=-1611 , STDEV=4
TCOUNT=9797 , STDEV=-110592
TBORLEN=12011 , STDEV=0
```

Figure 18 Sample Data Structure Listing

4 Segmentation

Image segmentation is a transformation from a multi-dimensional point by point (iconic) representation of an image to a representation of the image as a collection of regions which are homogeneous along some dimension. An object in the scene may be represented by one or more of these regions. Segmentation of a scene has little use by itself, but it is required before further symbolic analysis of the image can be attempted. The separate regions will be the basic units used in the symbolic analysis of the image. These will be the units used in feature extraction discussions in Chapter 5 and in further analysis of the image in Chapter 6.

We begin this chapter with a description of a basic segmentation procedure for use with multi-spectral images. We then introduce modifications to this procedure to reduce the time required for the segmentation operation, and to extend its usefulness to monochromatic images. The final section presents results for all of the images, with an evaluation of the accuracy of the segmentation and the time required.

4.1 Segmentation Method

The basic paradigm for the segmentation of images is the splitting of a region of the image into smaller regions, each of which are homogeneous in at least one spectral-based parameter. This basic technique was developed by Ohlander (1975). The operation of splitting a region is simply the application of a threshold on the feature values. The threshold limits are selected through the analysis of histograms of all features and the selection of a "good" peak. The use of histograms has long been used in computer analysis of images for the selection of the optimal threshold for the separation of various regions from the background (Prewitt, 1970).

It is easier to understand how this works by looking at a very simple example. For this example we will use an image that is blue on top and white on the bottom (one of the simplest two region natural scenes, Figure 1).

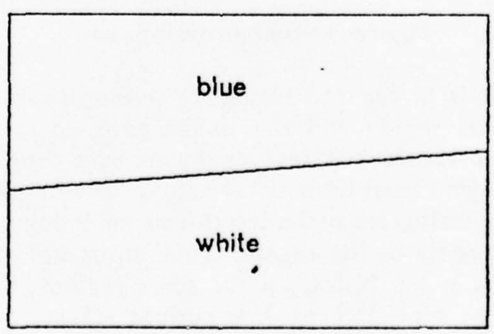


Figure 1 Simple Natural Scene

Ideally the histograms of the various features would show that all the points in the white region have one value and all the points in the blue region have another (or the same) value (Figure 2), but, generally, the noise in the image will cause the values to be distributed about the mean value of the feature (Figure 3). In this

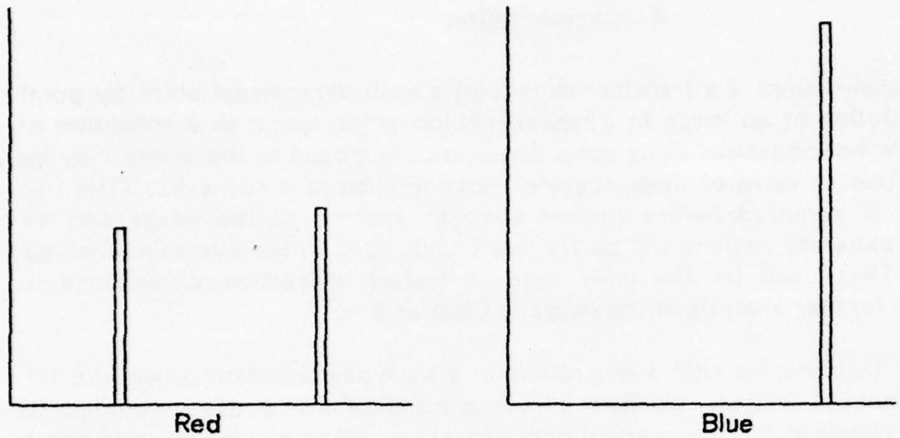


Figure 2 Histogram of Simple Natural Scene

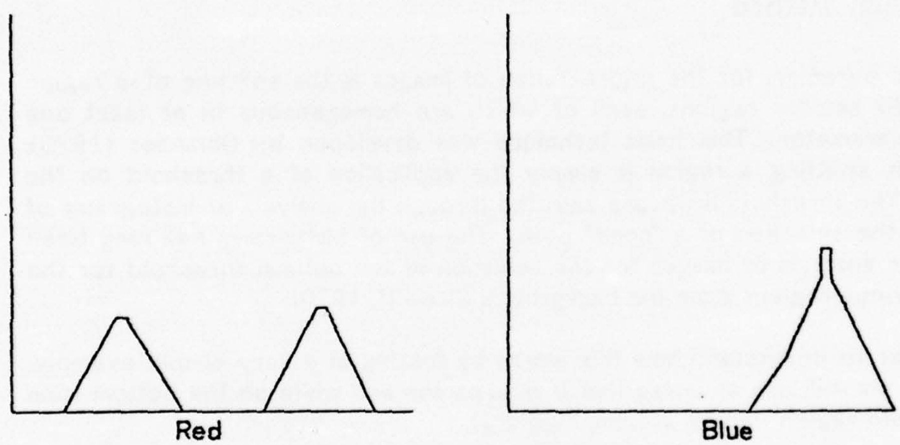


Figure 3 Modified Histogram

example the "good" peak is in the red histogram (either peak), with the lower peak corresponding to the blue region and the upper peak corresponding to the white region. This example also shows that feature values may overlap in one feature and not another (the white region must have equal values for red and blue, or it would not be white). The complete histogram of an image can be thought of as the sum of the histograms of all the segments of the region. Thus an image with two regions should have two separate peaks in the histogram for some feature, one with three regions should have three peaks, etc. But as the number of regions increases and the similarity of regions increases, the overlap of the peaks for the different regions also increases so that an individual peak in the complete histogram is really the sum of the peaks for several regions. As the number of regions increases even more, the valleys between peaks will be filled in by the values for these new regions.

In more detail, the segmentation procedure works as follows (see Figure 4 for a flow chart of the procedure):

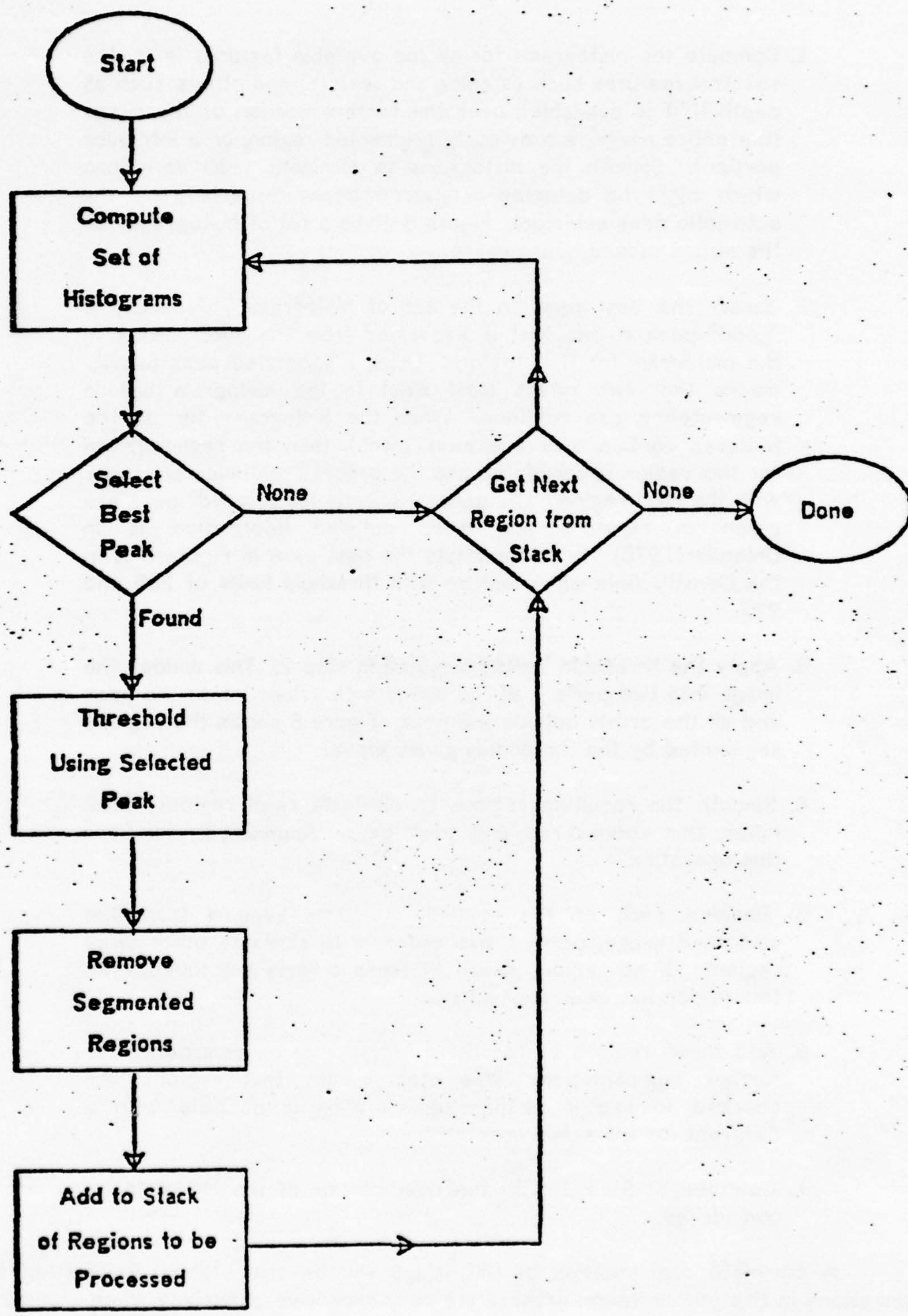


Figure 4 Segmentation Procedure Flowchart

1. Compute the histograms for all the available features (e. g. the spectral features such as color and texture, and others such as depth if it is available), over the current portion of the image (the entire image, a previously segmented region, or a left over portion). Smooth the histograms to eliminate small variations which might be detected as narrow peaks (especially by the automatic peak selector). Figure 5 gives a set of histograms for the entire second house image.

2. Select the best peak in the set of histograms. Generally a "good" peak is one that is separated from the other peaks in the histogram for that feature. Using a separated peak usually means that two peaks must exist in the histogram before segmentation can continue. When the histograms for all the features contain only one peak (each), then the segmentation for this region is completed and the process continues at step 1 with the next region. The general criteria for a "good" peak are given in Figure 7 and more detailed information is in Ohlander(1975). In this example the best peak in Figure 5 is in the Density (intensity) feature with threshold limits of 205 and 227.

3. Apply the threshold limits computed in step 2. This divides the image into two parts - all the points with values inside the peak and all the points outside the peak. Figure 6 shows the regions segmented by the thresholds given above.

4. Smooth the resulting regions to eliminate small regions, small holes, thin connections and small bays. Appendix 2 discusses this operation.

5. Remove each of the spatially separate regions from the smoothed image, using a size criterion to eliminate other small regions. If no regions which fit these criteria are found, then this region has been segmented.

6. Add these regions to the list of regions to be considered for further segmentation. This step implies that regions are checked to see if further segmentation is possible with a different (or the same) spectral band.

7. Continue at Step 1 with the next portion of the image to be considered.

A complete segmentation of this image will be given later. Several of the operations in this segmentation process are very expensive, especially when applied to very large pictures. One of these is the histogram computation which is applied to all the input parameters (in this case nine of them: red, green, blue, density, hue, saturation, Y, I, and Q). Appendix 3 tabulates the number of basic operations used per pixel for many of the segmentation operations. Other expensive operators are the refinement (smoothing) of the thresholded image and the removal of each of the

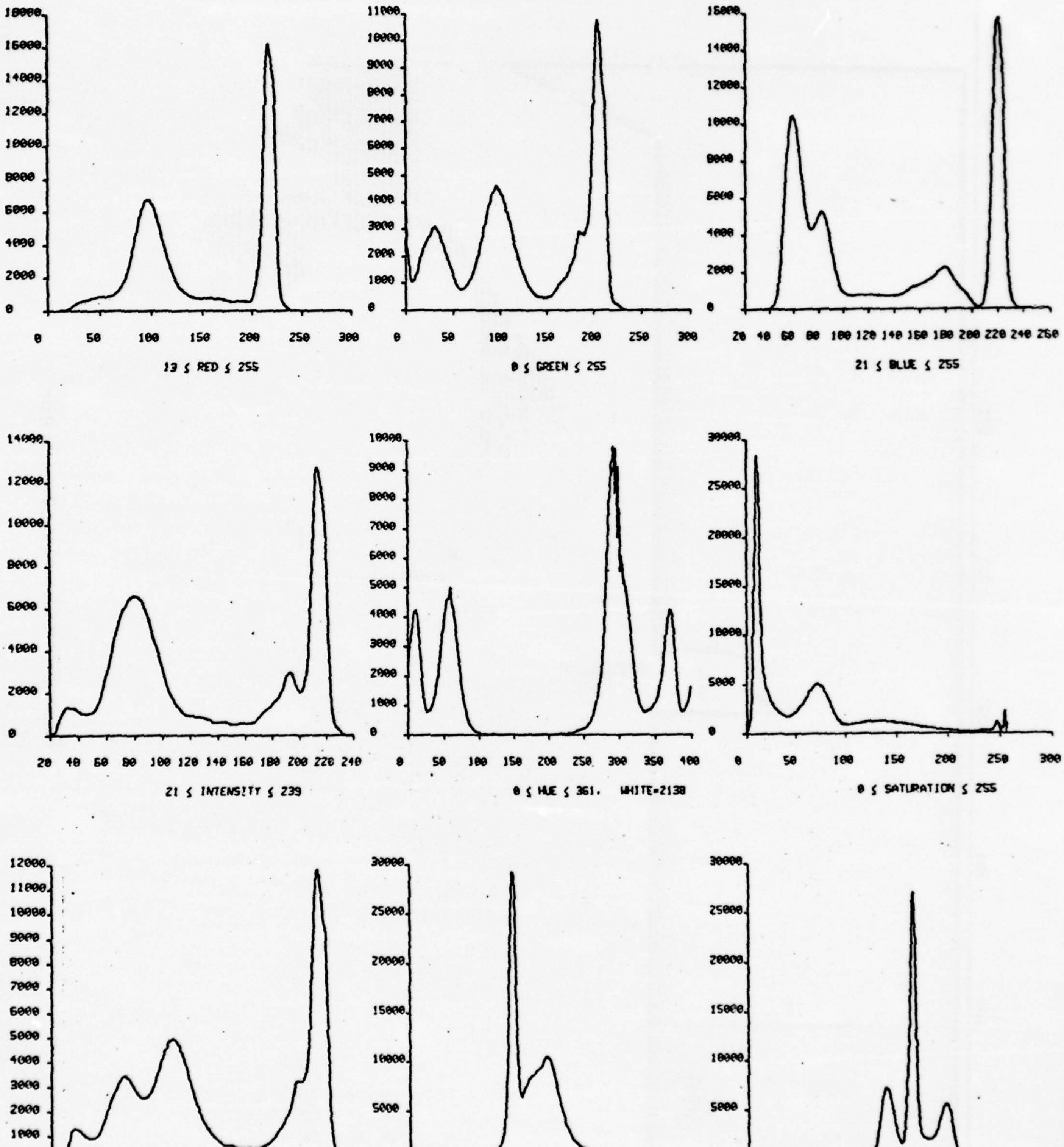


Figure 5 Histogram of Original Image of House-2

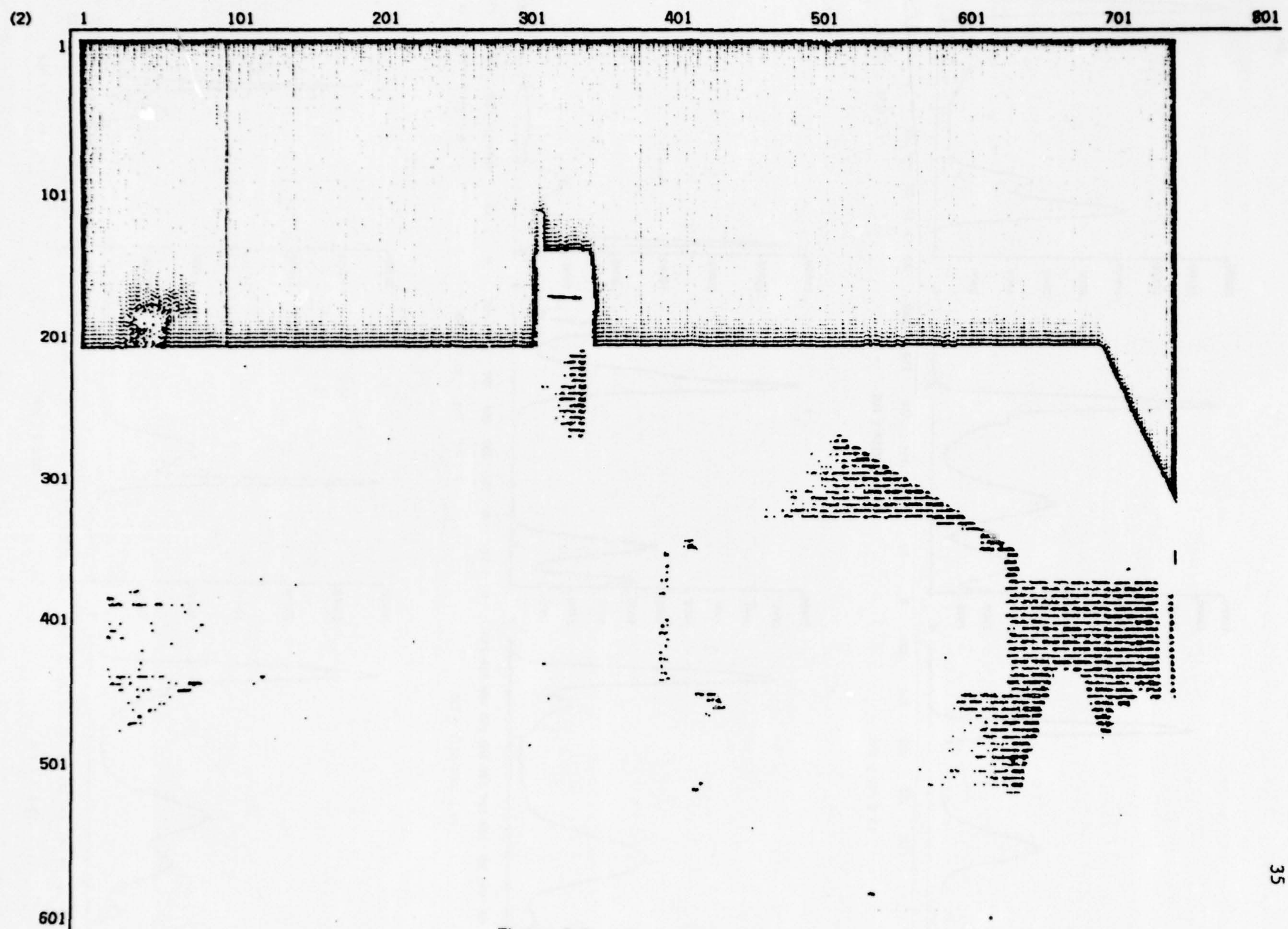


Figure 6 First Step of the Segmentation

0: Extreme intensity peak (bright or dark)
1: Very low minimum between two peaks, one larger than other
2: Less strict version of 1
3: Bimodal distribution
4: Peak in low saturation range (where applicable)
5: Single peak with large number of points in tail

Figure 7 Peak Precedence Criteria

regions. There is little chance to attain a significant speed-up of this process by merely modifying the programs, but there are modifications to the algorithm which offer substantial speed-ups due to a reduction in the use of these expensive operations.

4.2 Faster Segmentation

The path to a faster segmentation seems to be through the application of the expensive operators to smaller areas of the image. Several techniques offer potential savings:

1. Ordering of spectral bands by the likelihood of use in segmentation.

2. Selection of thresholds for the entire image based on histograms of a portion of the image.

3. Segmentation of important (large) regions using a reduced version of the image.

The first technique is applicable only to the segmentation of many similiar images. The experience gained through the segmentation of similiar images would allow the selection of the most likely spectral features (and possibly even thresholds) for several steps of the initial segmentation without analysis of all spectral features. This technique would require modification of the segmentation algorithm to look for a potential split in the more likely features, and to evaluate the other features if no divisions were located. This technique will not be explored further; it is only mentioned as one possible extension.

The second technique is feasible when a small area contains many representative regions. The regions also must be small with respect to the image, which is true of images taken a great distance from the scene such as satellite and aircraft images. An extended version of this technique will be discussed later in this chapter under the topic of monochromatic images.

The third technique can be applied to images which have relatively large regions. The regions must be large enough to be meaningful in the reduced image. This plan generation uses the same segmentation procedure as described above, and will be discussed next.

4.3 Segmentation by Planning

Planning consists of the reduction of a problem to a manageable size, the generation of an approximate solution to the original problem (a plan), and the extension of the plan to an accurate solution of the original problem. In computer vision the scale of the problem is usually reduced by finding the solution in a smaller image.

The human visual system uses a type of planning in determining what to look at. Since the receptors of the eyes are concentrated in one area (the fovea) the eye must be directed to interesting areas by the gross level processing in the periphery.

Kelly (1970) applied planning to the analysis of pictures of human faces. By using reduced images, his programs were able to find the outline of the head by searching the image and by using backup when errors were found. This approximate outline was then used as a guide to locate the outline of the head in the full size image.

Hanson et al.(1974, 1975) are working on an image analysis system in which most of the image processing involves the application of an operator which reduces the size of the image (by a factor of two) or the application of an operator to project information gathered (or regions segmented) on a reduced image back onto the larger image. The step by step reduction and processing causes plans to be generated in a reduced image which can be used to guide processing in the larger image.

A set of reduced images can be used to generate a plan for the segmentation of the full size image. At worst the plan will contain only the large, clear, and maybe important regions. The procedure for segmentation which was described above can be used with few modifications.

The planning process can be extended to many levels of reduction (as is used by Hanson et al.), but our use of planning will be limited to one level, usually a reduction by eight and sometimes by four. The same segmentation procedure is used on the planning images as was described above for full size images.

4.3.1 Plan Generation Results

We applied this planning technique to generate a plan for the four images in the house and cityscape scenes. In this subsection we will give a detailed discussion of only one of these images (the second house image), and will present the segmentation of the other three images at the end of this chapter. With some modifications which will be discussed later, this planning procedure was also used for the other scenes.

We reduced the original red, green, and blue parameter images by a factor of eight in each direction (the total size was thus reduced by 64). The amount of reduction depends on several factors including: the size of the desired regions (the region must be large enough to be extracted in the plan), and the total image size (it is desirable that the reduced image be small enough to completely fit in primary memory,

i.e. at most about sixty thousand words are available for images). The reduction program gave more weight to the points in the center of the window than to points on the edge, and produced a variance image in addition to the mean image. The center of the window is weighted more heavily than the outside as a compromise between reduction by sampling and reduction by averaging. The weights are computed as $2^{-(\text{distance from the center})}$. Figure 8 gives the weighting values which were used. The weights are scaled to make the mean and variance computation easier (the values in the figure are rounded). The other color parameters (Density, Hue, Saturation, Y, I, and Q) were then computed from the reduced images (see Chapter 5 for a definition of these features). Each reduction operation for the house scene (one operation for each color of the three color image) requires about 78.33 million operations (about 140 operations per pixel of the original image) for a total of about 234.99 million operations to reduce all three colors.

.003	.005	.007	.008	.008	.007	.005	.003
.005	.008	.0125	.016	.016	.0125	.008	.005
.007	.0125	.022	.0315	.0315	.022	.0125	.007
.008	.016	.0315	.058	.058	.0315	.016	.008
.008	.016	.0315	.058	.058	.0315	.016	.008
.007	.0125	.022	.0315	.0315	.022	.0125	.007
.005	.008	.0125	.016	.016	.0125	.008	.005
.003	.005	.007	.008	.008	.007	.005	.003

Figure 8 Weights for the Reduction Program

The plan generation procedure started by segmenting the bright intensity peak from 205 to 227 (Figure 9). This selected the sky region above the house (Figure 10). The next peak is also in the intensity parameter from 24 to 51 (the dark peak) (Figure 11); this segmented some of the bushes in front of the house (Figure 12). The next peak was in the Red parameter from 62 to 131 (Figure 13); this selected the roof, lawn, window, and door areas (Figure 14). This continues for several more steps until the image is segmented. Most of the regions are completely segmented on the first pass and do not require further segmentation. One of the regions that required further segmentation was the lawn area which was segmented on the third iteration. Even though red was used in the original segmentation, it is not used in this second segmentation (Figure 15); the best peak is in the Q parameter from 220 to 260. The complete plan for the house is given in Figure 16. There are 21 basic regions in the plan (plus eight which were segmented further). The histogram peak selection was used to find a split nine different times.

4.3.1.1 Plan Timing

Figure 17 gives the timing summary for the plan generation (in millions of operations) of the house scene. The total computer time was a little more than two minutes (the real time was about 54 minutes, and included time for graphical displays and saving intermediate results). These times were summarized from the computer generated timing files and do not include some of the times for overhead operations or the times for operators that required much less than one percent of the total time (the

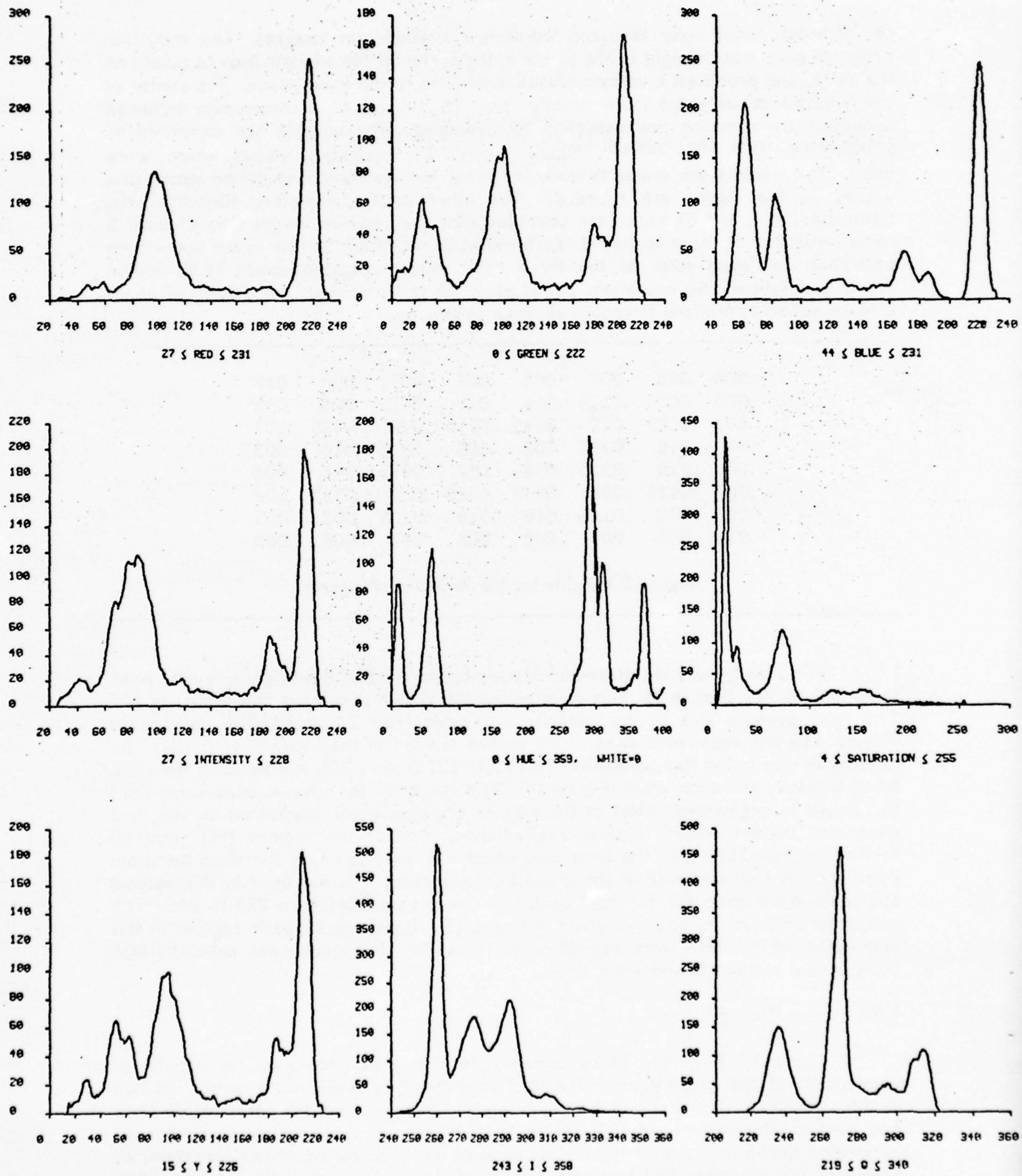


Figure 9 Histogram of House-2 Complete Image

Figure 10 Regions Extracted Using Density from 205 to 227

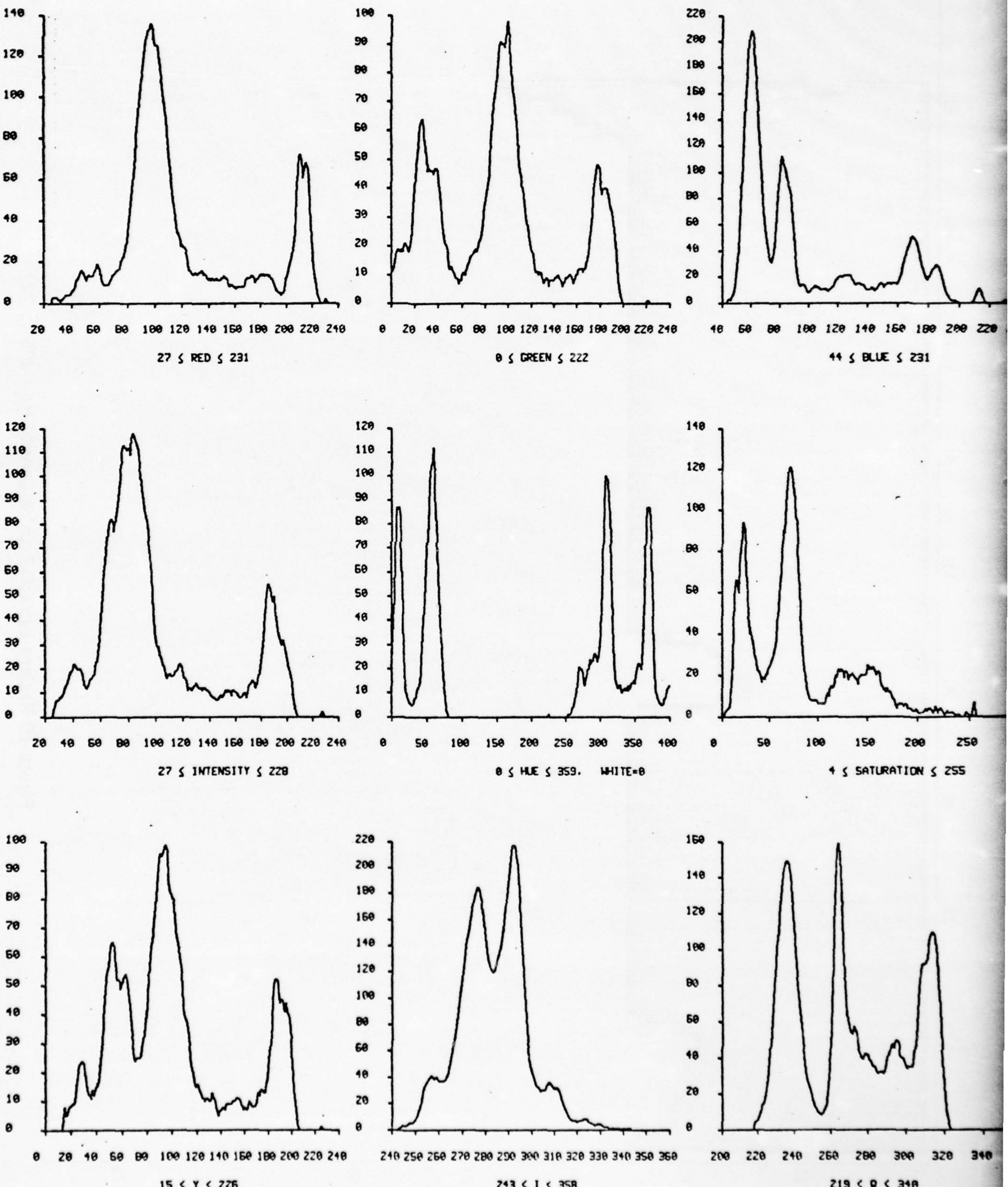


Figure 11 Histogram of House for the Second Segmentation Step

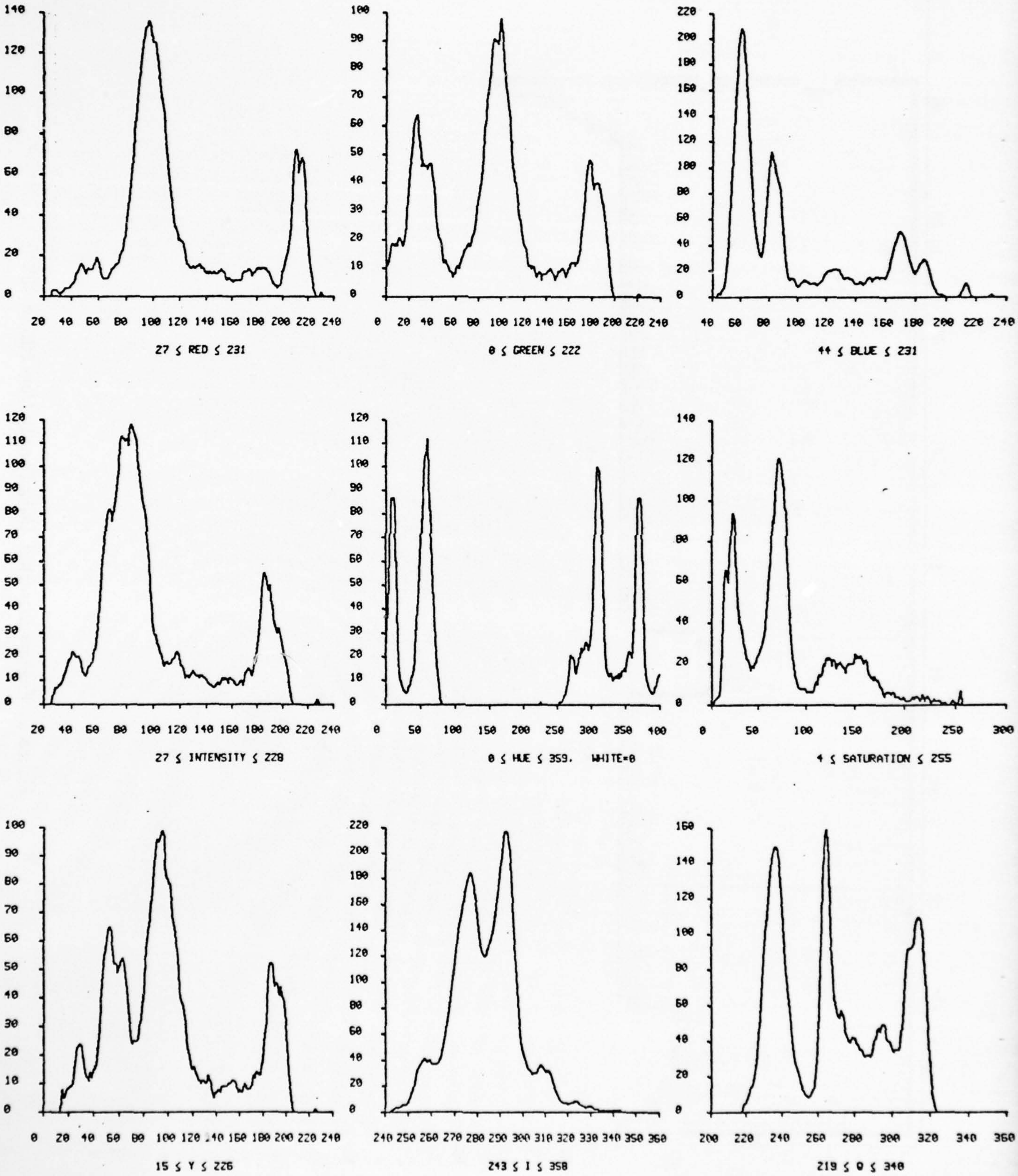


Figure 11 Histogram of House for the Second Segmentation Step

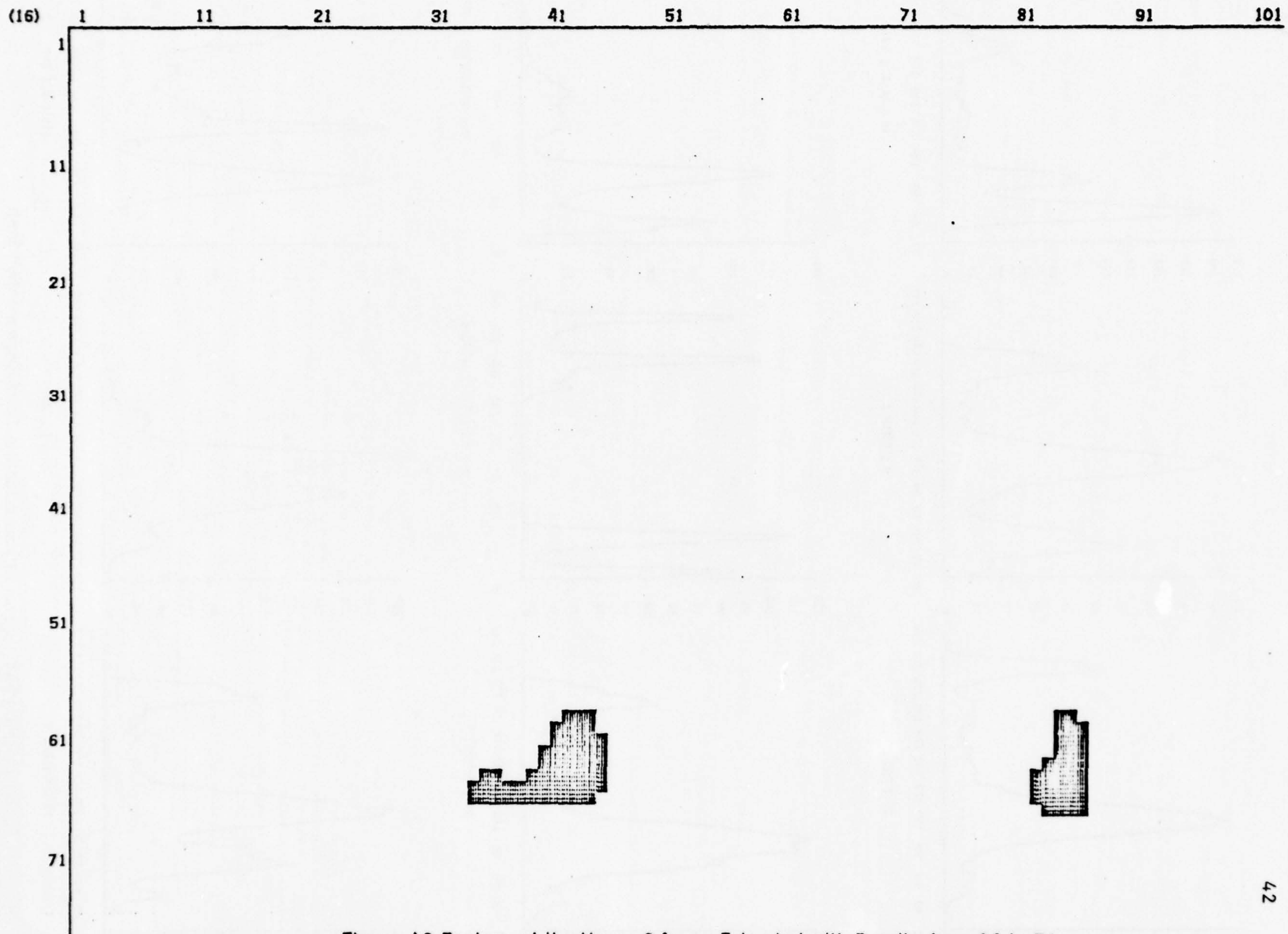


Figure 12 Regions of the House-2 Image Extracted with Density from 24 to 51

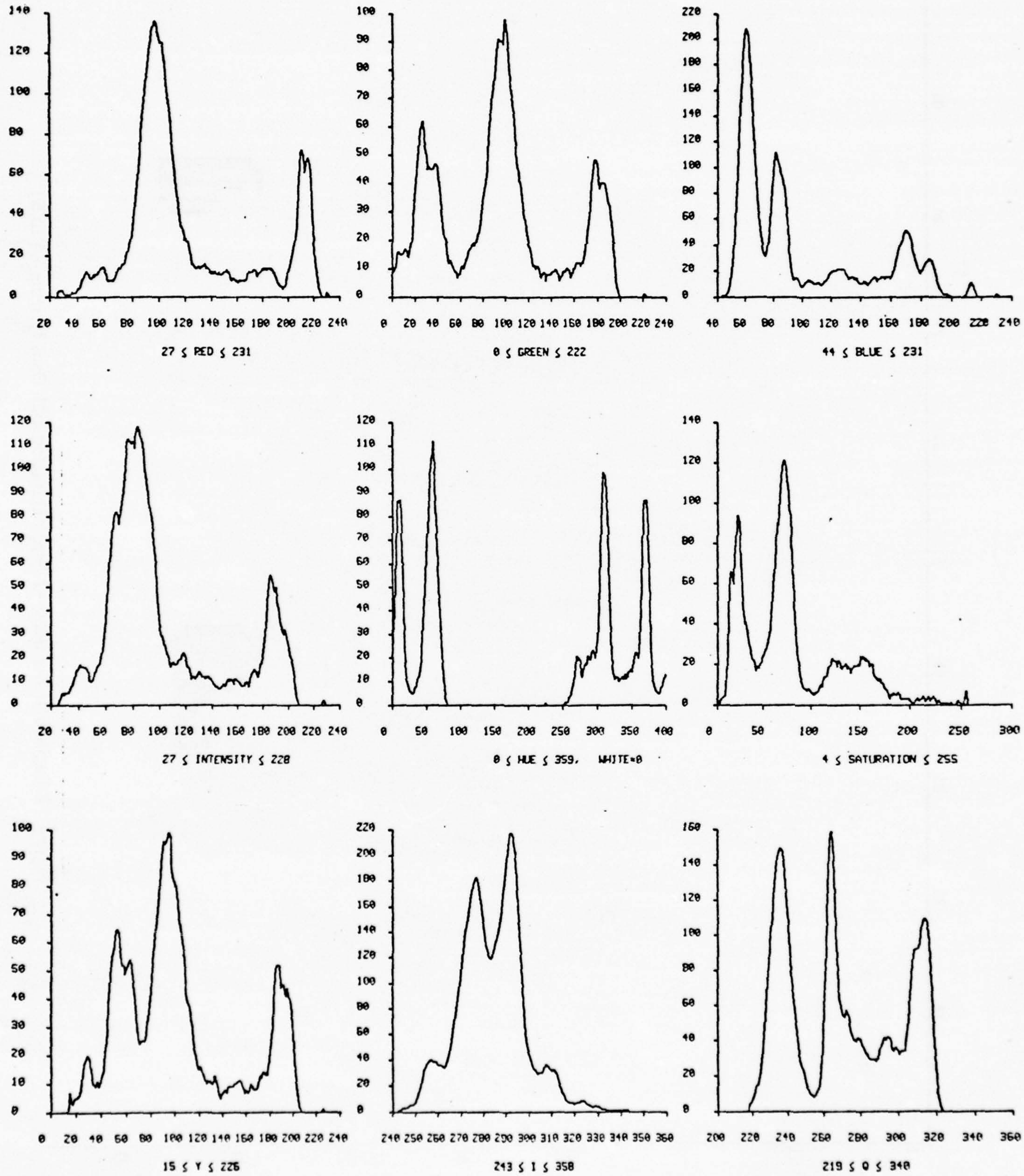


Figure 13 Histogram of House for the Third Segmentation Step

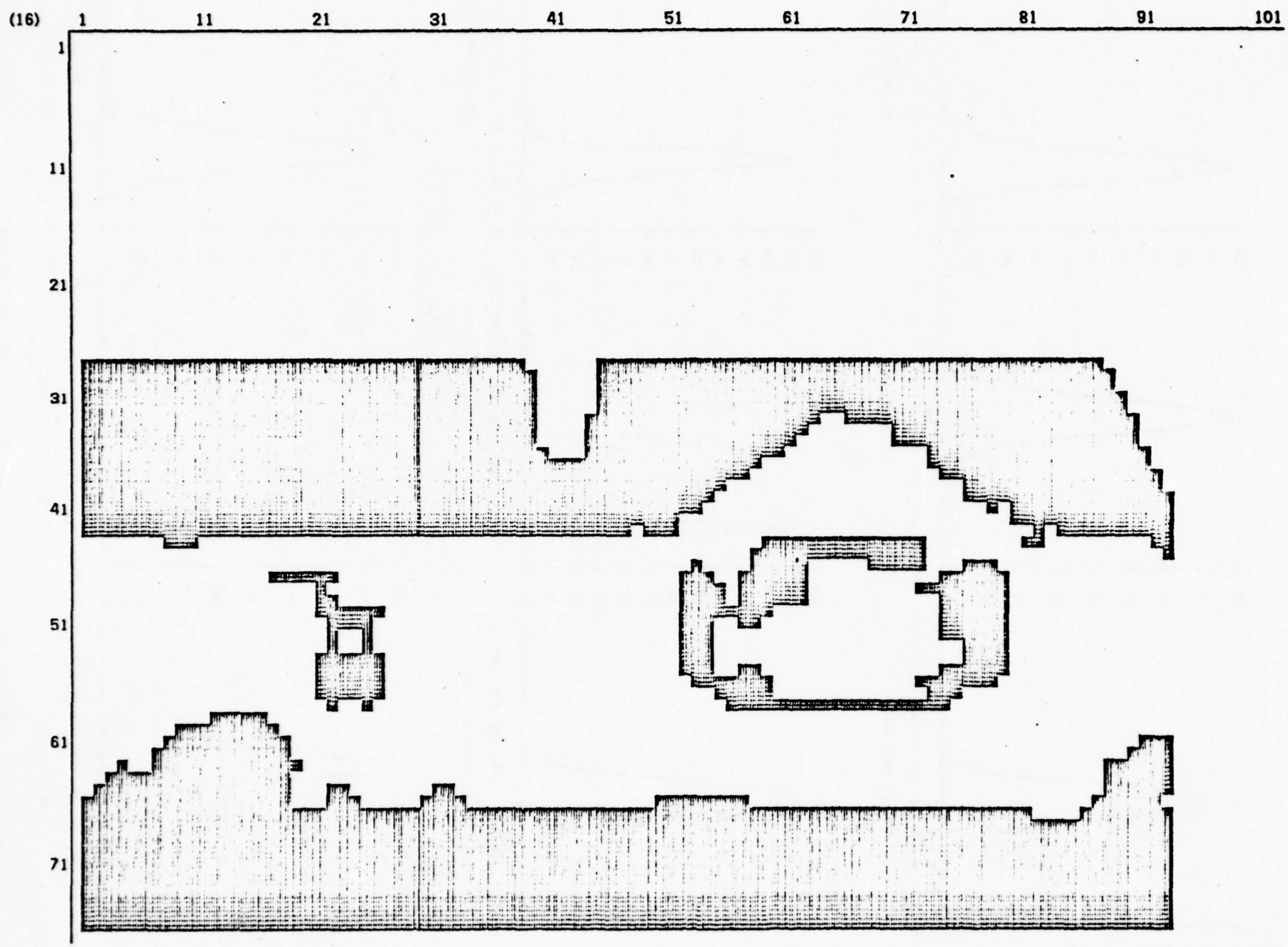


Figure 14 Regions of the House-2 Image Extracted Using Red from 62 to 131

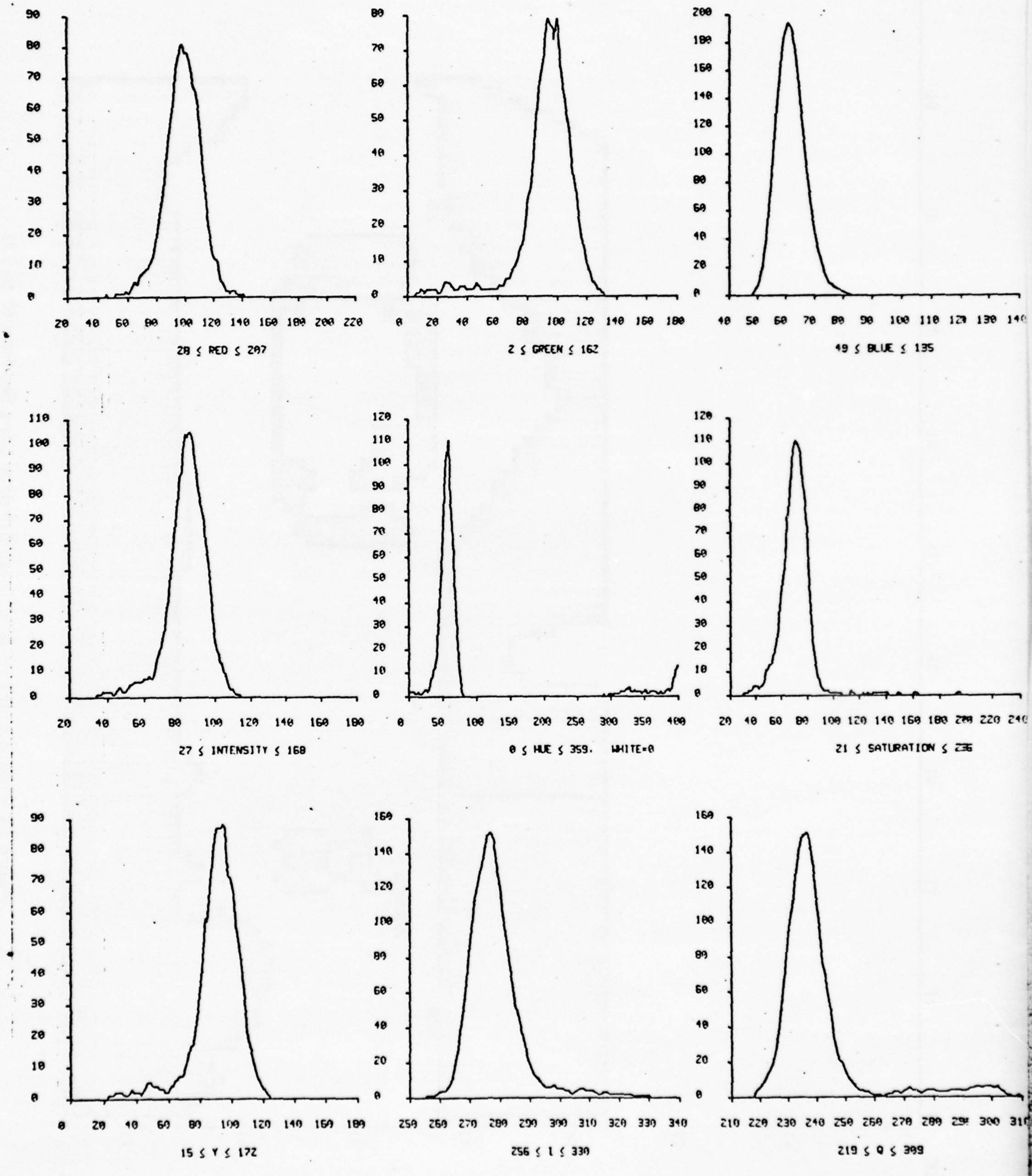


Figure 15 Histogram of Lawn Region of House-2 Image

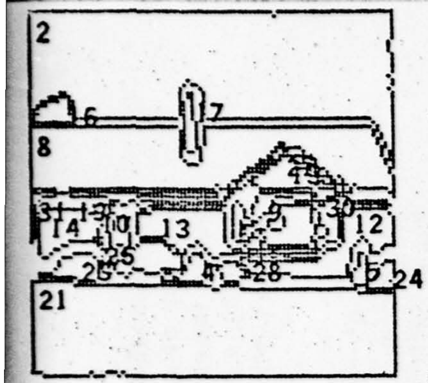


Figure 16 Complete Plan for the Segmentation of the House-2 Image

timing overhead is normally less than 2%). As can be seen from this summary, the most expensive processing operation is the histogram generation which takes about 56% of the time, but only half of this processing depends on the picture size; the other half is histogram array processing (smoothing the array) and depends on the size of the pixels (byte size). The peak selection takes about 20% and also depends only on the number and range of the parameters. About 23% of the time is consumed by steps 3 to 5 (threshold, smooth, and region extraction) of the segmentation process. The times for these operations are dependent on the picture size.

Operation	Millions of Operations	Percent of Total	Number of Times Used
Histogram Computation			
Generation of array	10.64	26.7	117
Smooth array	11.37	28.5	117
Other	0.29	.7	13
Peak Selection	8.18	20.5	13
Threshold	1.38	3.4	13
Smooth	3.27	8.2	13
Region Selection			
Initialize	3.28	8.2	13
Select a region	0.61	1.5	22
Save masks	0.83	2.1	32
Total	39.84	--	

Figure 17 Timing Summary for Plan Generation House 2

4.3.1.2 Plan Evaluation

The plan generation was intended to segment the major (large) regions in the scene, which it does well. The time for the plan generation alone is significantly less than the time for a complete segmentation of a full size image.

Because of the smoothing of the image in the reduction operation, many of the textured regions in the full size image will be relatively homogeneous in the planning image. Thus we are not confronted with some of the problems that such "busy" regions caused Ohlander.

The smaller images caused some problems in the region splitting analysis. The regions that were extracted could be small (35 pixels or more), so that the histogram of such a segmented region could have many false peaks because the values in the region are scattered throughout the entire range of the threshold used for the extraction, which could easily cover 35 or more different values. This degeneracy of

histograms also occurs during the segmentation of the whole image, so that many regions are left as unsegmented areas of the image, after the application of the plan generation.

4.3.2 Expansion

When a plan is generated there must be a method for transforming the plan into a segmentation of the full size image. An approximation of the full size region can be generated by expanding the plan generated mask by the *reduction factor*, but this will not produce an accurate segmentation. Therefore, the expanded segmentation mask must be refined by using the same threshold parameters which were used to generate the plan mask. The following procedure has been implemented for this purpose. Figure 18 gives a flow chart for this procedure.

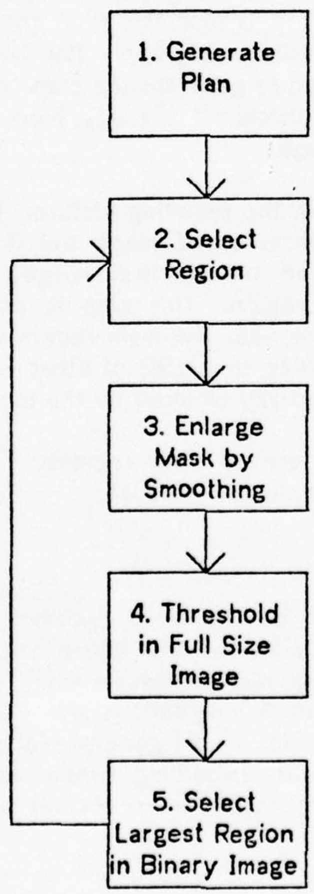


Figure 18 Expansion Flow Chart

1. Begin with the plan - a partial or complete segmentation of the reduced scene. Figure 16 shows the outlines of the regions in the plan (the second house image).

2. Select the next region in the plan (starting with the first region) which

is not further divided into smaller regions (i.e. a region that has no descendants). Figure 19 shows such a region (the scale is selected so that this figure shows the region the same size as it will be in the full size version); this is region number 8, the roof, in the plan.

3. Enlarge the binary image (mask) in the plan by adding a layer of pixels (of "1"s) on the outside of the region. This is necessary to allow for the nonexact alignment of the plan region with the full size region. Figure 20 gives the enlarged mask. This enlarging is done by the smoothing operator as discussed in Appendix 2.

4. Expand the plan mask by the same factor that was used in the reduction of the image. Thus, if the reduction factor is eight, then each point in the plan mask is duplicated 8 by 8 (i.e. 64) times in the expanded mask. This mask is not the final result; it needs to be refined. Using this expanded mask, apply the same threshold to the full size image as was used to generate the plan. Figure 21 shows the results of applying this threshold (Density from 62 to 131) to the applicable area of the image.

5. Select the largest region in the resulting picture. Many times there is only one region in the thresholded image, but if a second region is relatively large (compared to the first image), it should also be retained as a separate region. This step is primarily intended to eliminate the small regions near the main region which may have the same spectral characteristics, or pieces of other (large) regions which are near enough to be partially covered by the expanded mask.

6. Continue at 2 until there are no more regions. Figure 22 shows the final expansion of the regions in Figure 16.

4.3.2.1 Expansion Timing

Figure 23 gives a summary for the times required for the generation of an expanded segmentation from the plan for the house segmentation. As would be expected, the operations on the large masks consume most of the time. The smoothing operations are applied to remove small indentations and small regions from the mask. This operation could be eliminated; this would generally affect only the shape of the resulting regions. Elimination of the smoothing would also mean that the region extraction procedure would need to expect more regions and thus might need more temporary space.

4.3.3 Overall Segmentation Times

The plan generation and expansion operations, combined, take about 12.5 minutes of computer time. This is equal to about 226 million operations for the segmentation of a picture with .5 million pixels (with the 9 parameters it is about 4.5 million total pixels) or about 450 operations per pixel.

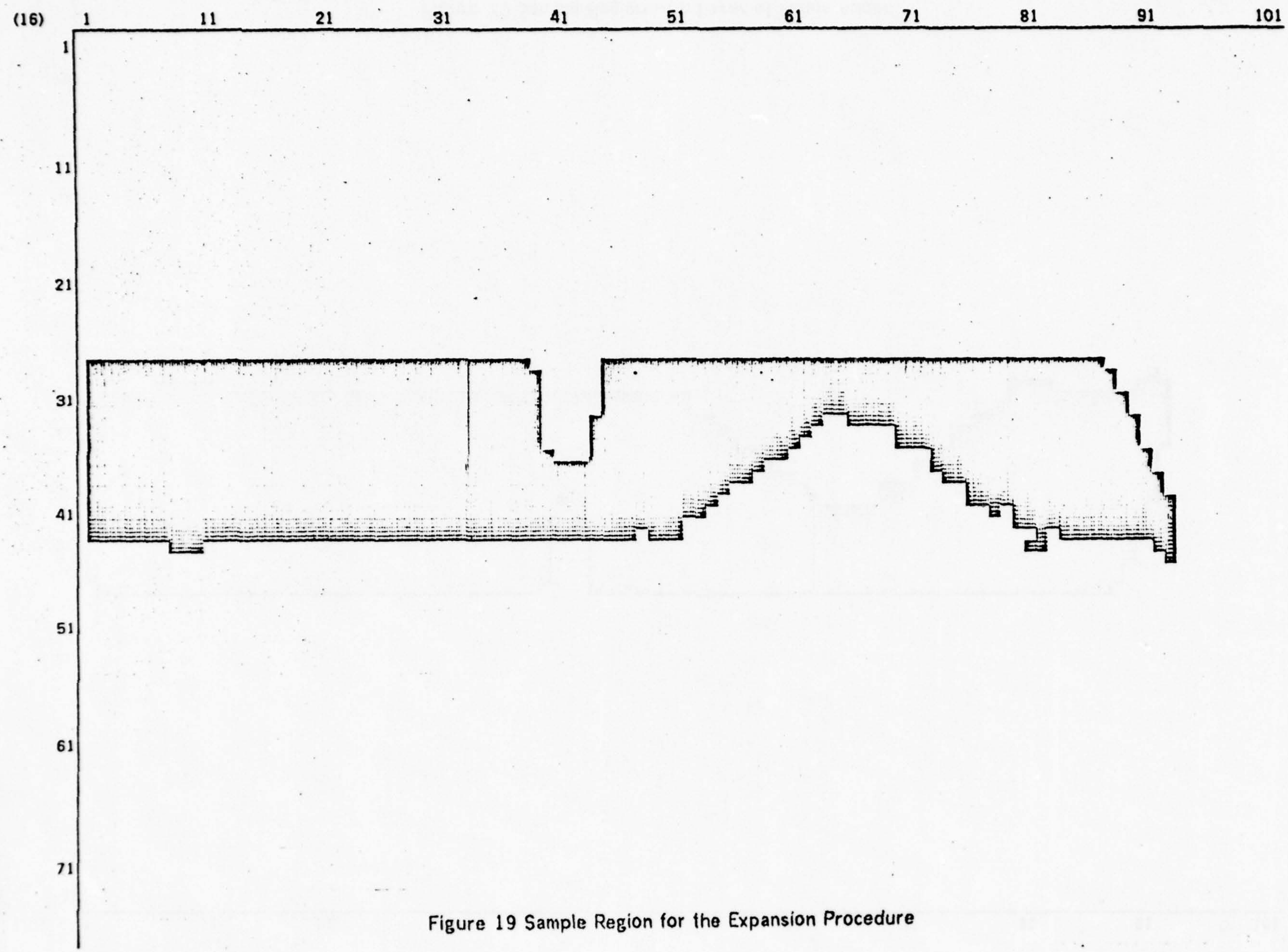


Figure 19 Sample Region for the Expansion Procedure

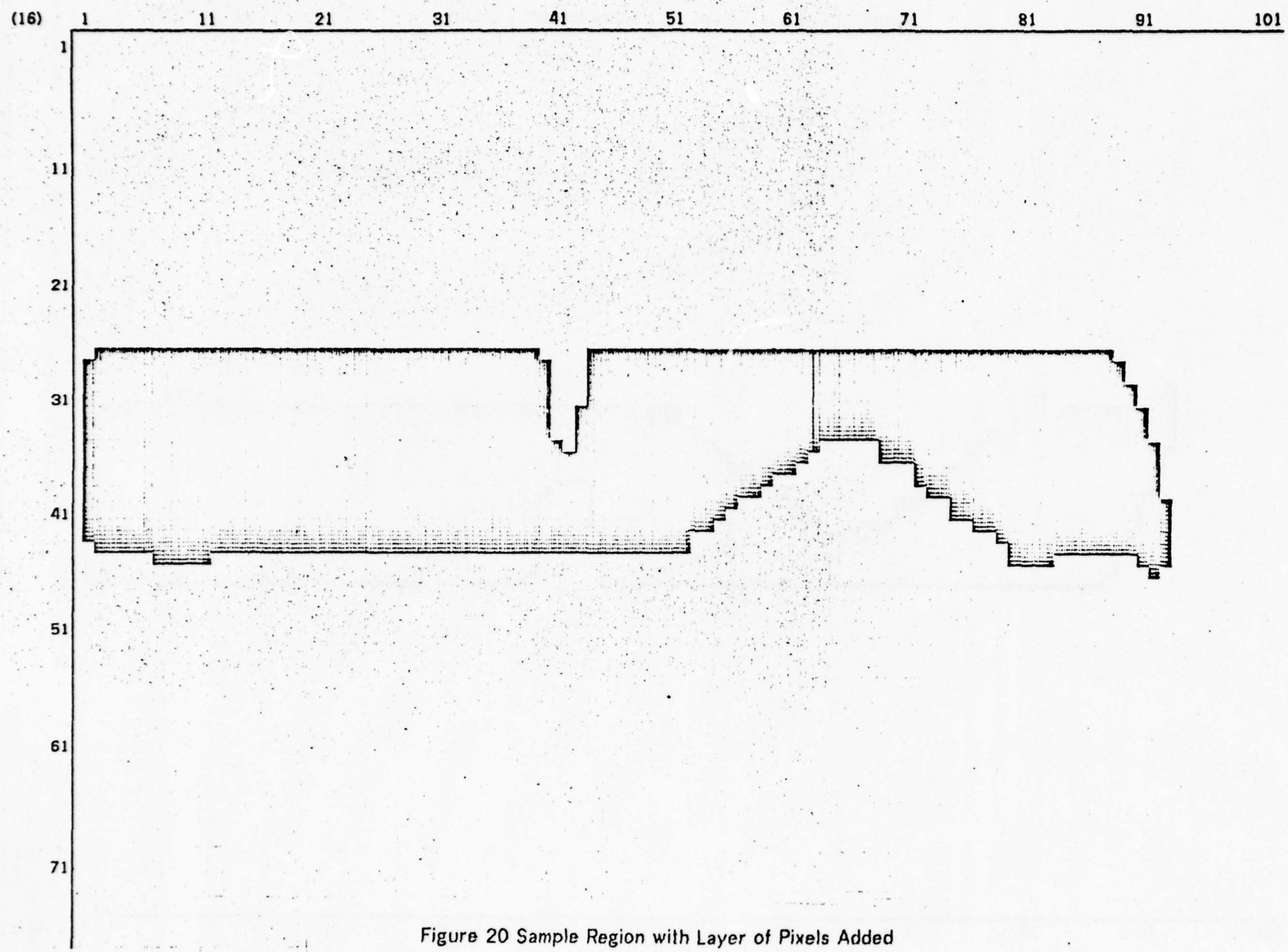


Figure 20 Sample Region with Layer of Pixels Added

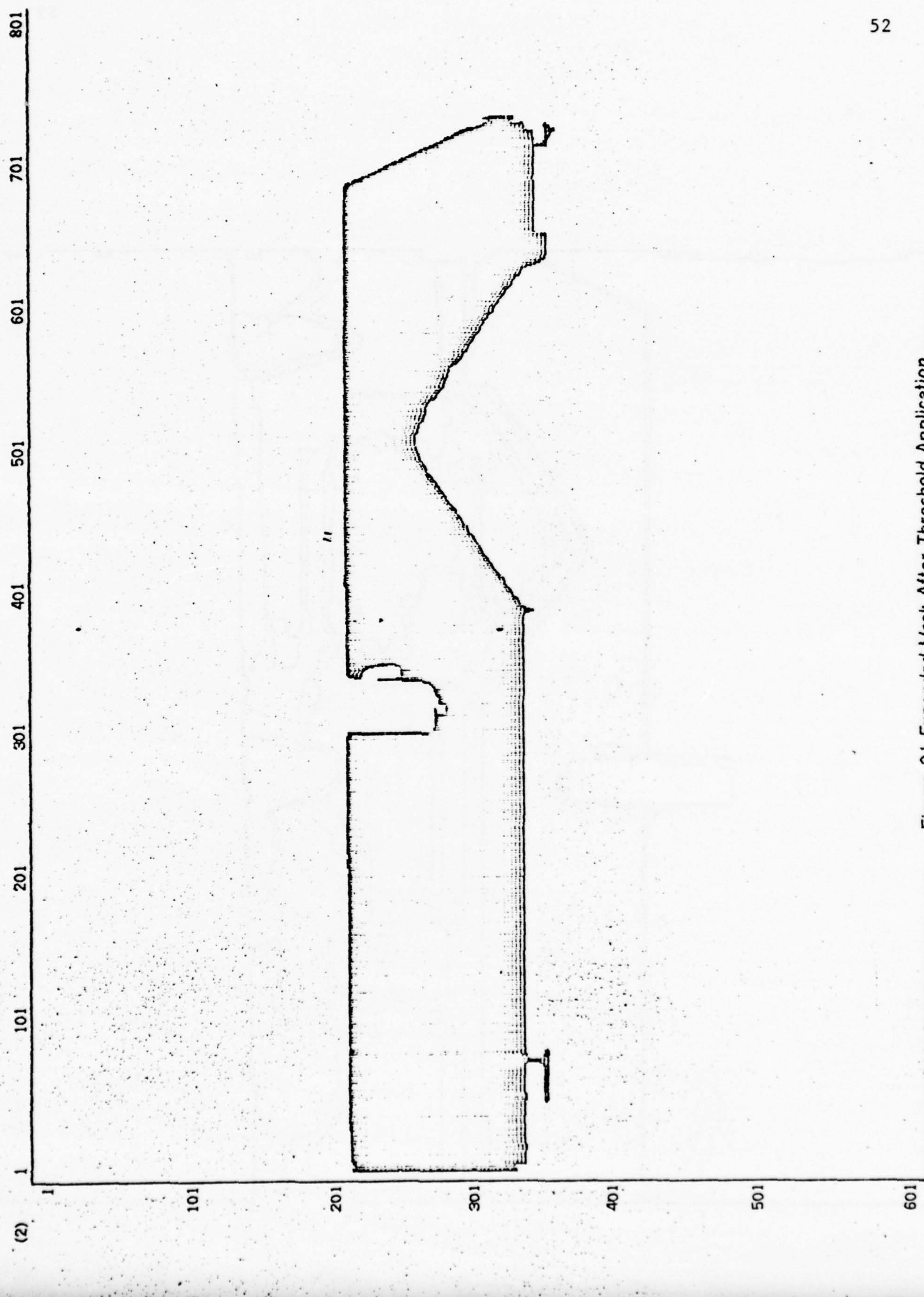


Figure 21 Expanded Mask After Threshold Application

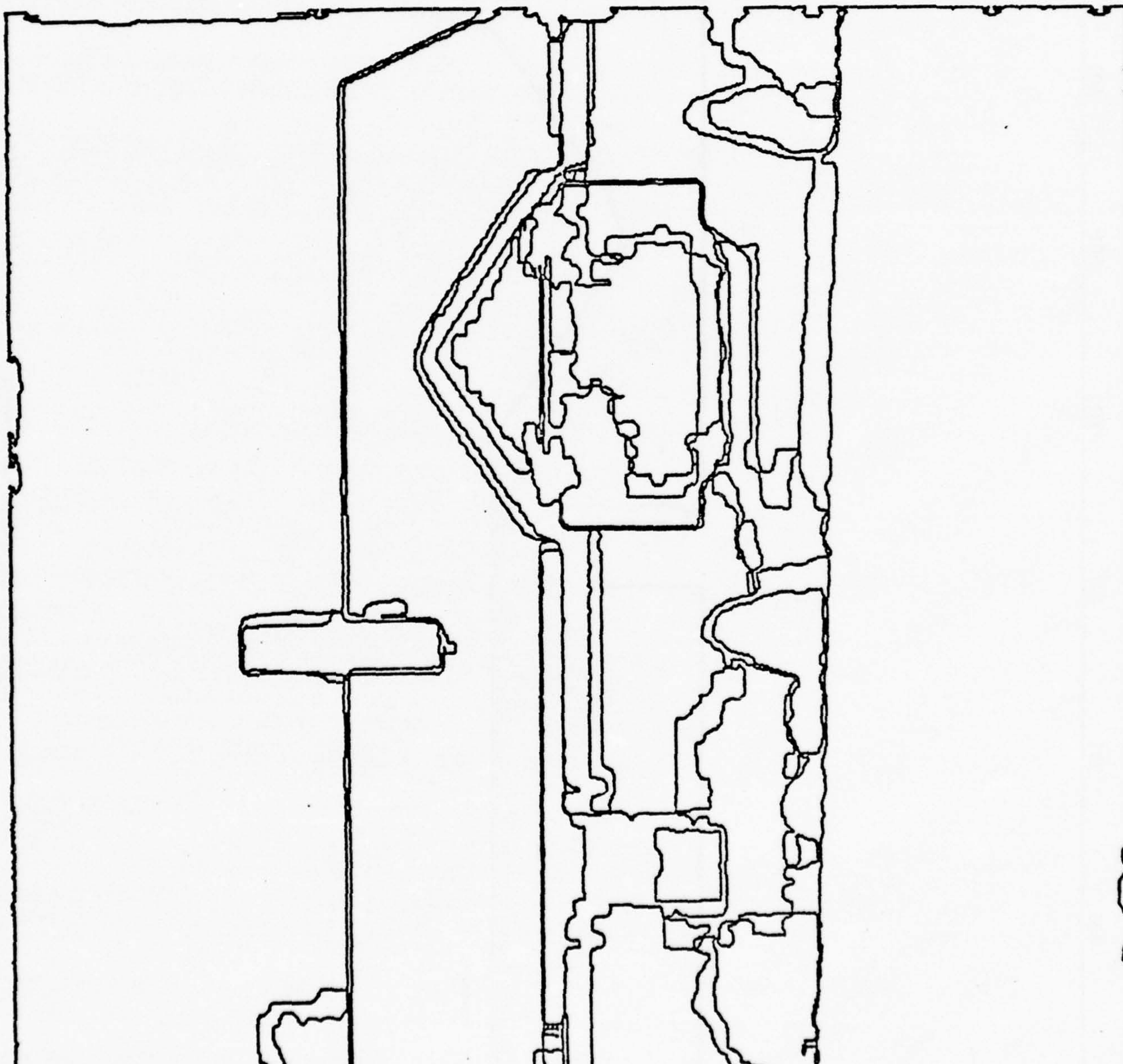


Figure 22 Complete Segmentation Results for House-2 Image

Operation	Millions of Operations	Percent of Total	Number of Times Used
Enlarge Small Mask	1.19	.6	21
Large Mask Threshold	23.44	12.6	21
Smooth Large Mask	116.72	62.7	63
Extract Regions	44.92	24.1	21
Total	186.28		

Figure 23 Expansion Timing House 2

4.4 Using Knowledge in Segmentation

The segmentation procedure as described so far attempts to completely segment the image without relying on outside knowledge. In a general image understanding system, this complete segmentation would rarely be required as the first step. Generally, the extraction of several large general regions, or regions with certain characteristics, or the continuing of the segmentation of large general regions is more important than the generation of a complete initial segmentation. The segmentation procedure we described above can be used for this type of partial segmentation with very few modifications; those would be in the outer level control of the procedure. Segmentation based on specific characteristics requires an alteration of the peak selection procedure to look for the specific peak and no other (e.g. only bright peaks in red, the biggest peak, etc.). When looking for a specific peak the constraints on the "goodness" of the peak for acceptance may be relaxed. Large general regions are extracted by applying the basic segmentation procedure (possibly with altered peak selection priorities), but eliminating the requirement that all regions must be checked by the segmentation procedure for further segmentation. The first two scenes (the house and the cityscape) did not require any of these modifications because a near complete segmentation was desired, but the other scenes, as will be seen, use specific knowledge about the task to determine how the segmentation will proceed.

For example, in both of the LANDSAT images (Figure 3.6 and Figure 3.7) the task, for the segmentation step, is to locate (i.e. segment) several lakes in the two images. The outside knowledge also describes the spectral characteristics of these lakes as the darkest regions in the fourth spectral band since the water absorbs the infra-red radiation. Given this knowledge, it is necessary only to compute the histogram of the band four data and to determine the upper threshold of the peak at (or near) the zero intensity level. The use of this knowledge means that it is necessary to compute only one histogram and there is no need to analyze any of the other three spectral bands. For another task, one might use another band and another peak might be specified.

In this scene, the peak was located at the zero intensity level, with the upper threshold given by the minimum between this peak and the large peak of medium intensity values. This minimum value occurs at about 6 (this histogram is given in Figure 24). As before, the segmentation procedure is applied to a reduced image (hence a plan is generated), rather than to the full size image so that the small shadows are not segmented. The expansion procedure, described above, is used to expand the plan regions. The lakes (and other regions) in the full scale image are shown in Figures 25 and 26. The two large lakes near the left edge are segmented in both images along with several long thin lakes near the center of the images (below the white snow area). In the first image, another large lake (above the snow region) is also segmented, but it is obscured by clouds and is not segmented.

The LANDSAT task also requires the location of snow cover regions, which are defined as the large high intensity regions in the image. The bright regions are given by threshold limits of 34 and 63 (in the fourth band, see Figure 24). The larger regions generated with this threshold are shown in Figures 27 and 28 (the plan) and Figures 29 and 30 for the full size segmentation (which includes the lakes).

The use of partial segmentations will be very important for the matching and change analysis discussed in Chapter 6 since many of the less important (less likely to match) regions are eliminated from the analysis by the simple process of never generating them.

4.5 Segmentation of Monochromatic Images

When a segmentation procedure has been developed for one type of image, it is usually not the case that the procedure will work on very different types of images. The region growing system of Yakimovsky(1973) was applied to two very different types of data by using a different world model for each type of image (outdoors and heart angiograms). But, this system required a learning phase to generate the world model for the different scenes. Our segmentation procedure was originally developed for images of natural scenes with many spectral bands, large regions, and oblique views, and might not be expected to work very well on monochromatic aerial images. In scenes with many small different objects (as is the case with aerial photographs), the histogram will generally have only one peak because the range of intensities for each object will probably overlap with the ranges for other objects. Because there are no other spectral inputs that are directly available, we can not combine several inputs to generate other spectral features (such as is done to get Y, I, and Q from red, green, and blue).

As expected, when the original procedure is applied to the large black and white aerial images, a "good" segmentation is limited to partial segmentations of the type discussed above (e.g. the very bright or very dark regions or large varied regions), but even these peaks for the partial segmentation may be obscured. Thus our monochromatic images introduce two new problems. First, there are too few spectral bands for adequate separation (i.e. only one). Secondly, there are too many small regions which cause all the separate peaks to blur into one.

We attack the first problem by the introduction of simple "textural" measures which can be used to generate the reduced image instead of the simple weighted

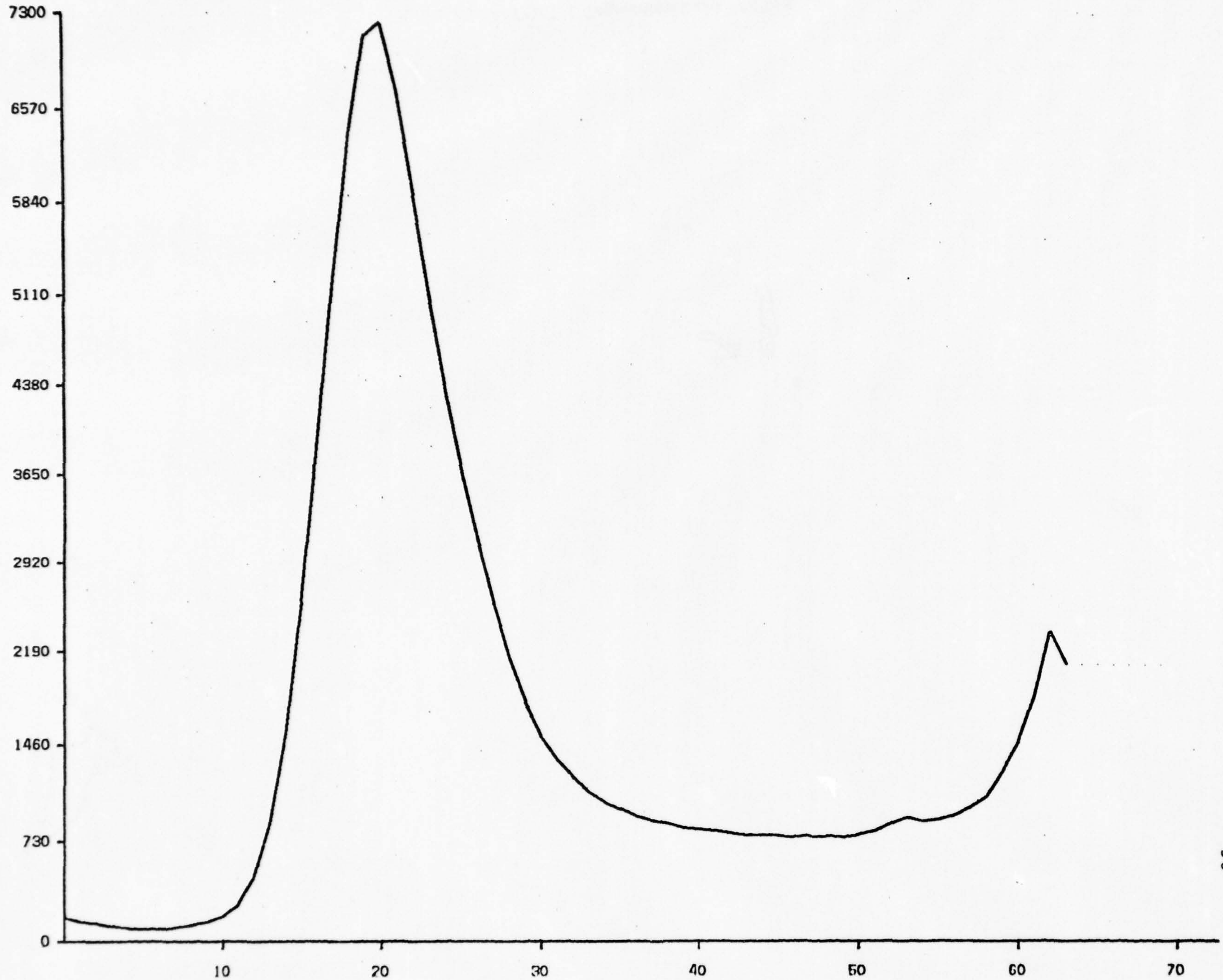


Figure 24 LANDSAT 1 Histogram of Band 4

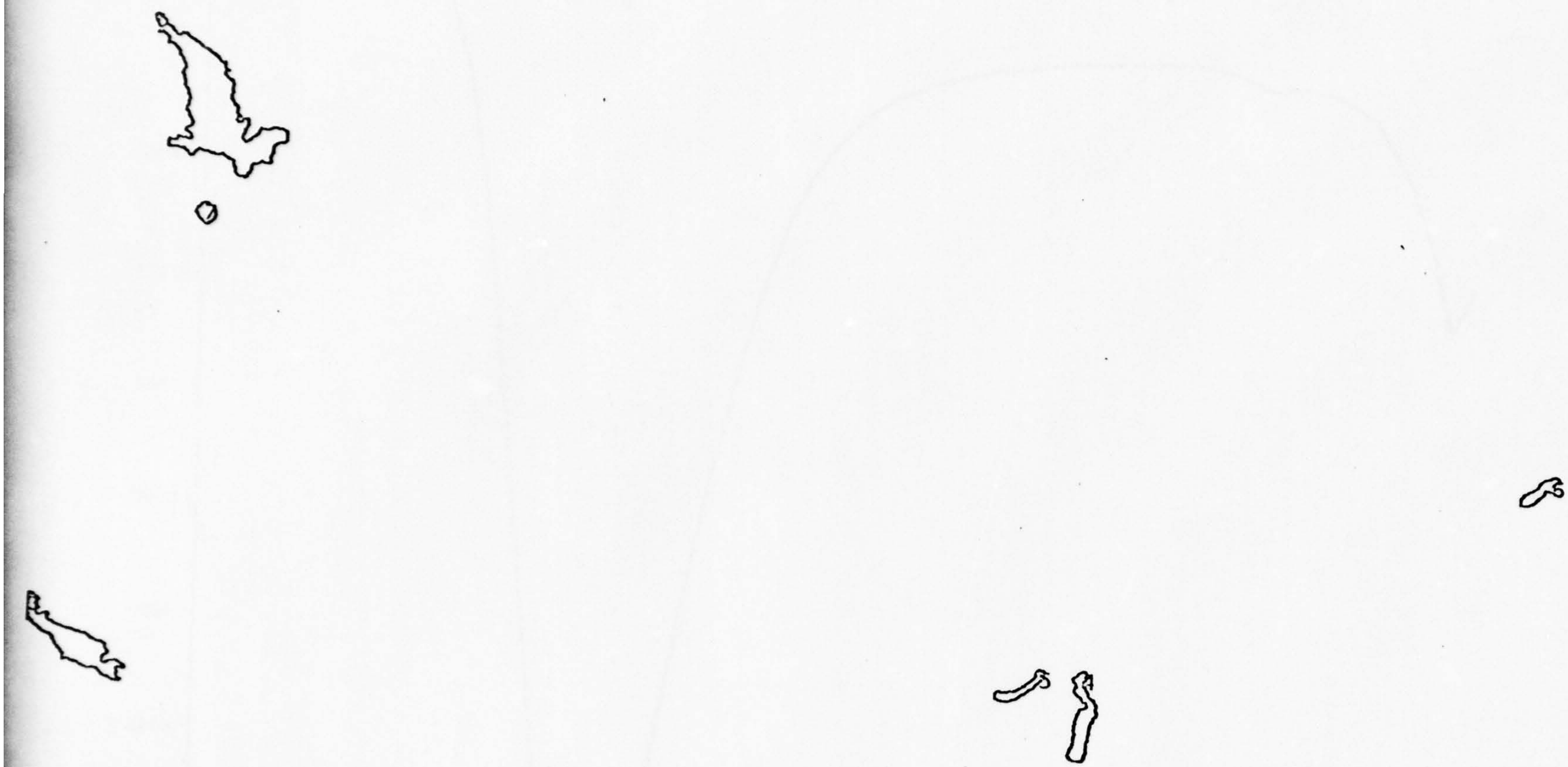


Figure 25 LANDSAT 1 Segmented Lake Regions

Figure 26 LANDSAT 2 Segmented Lake Regions

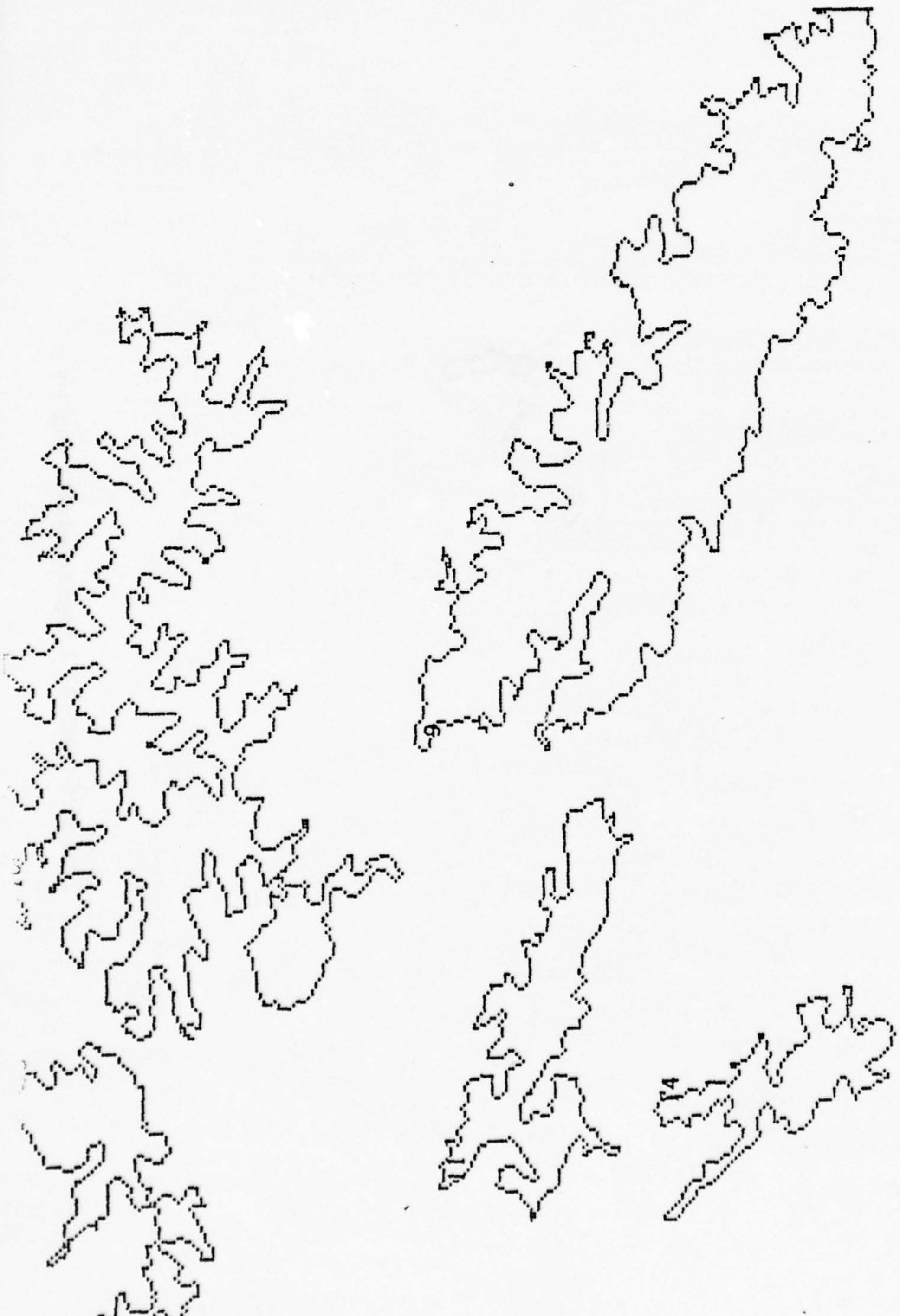


Figure 27 LANDSAT 1 Plan of Snow Regions

Figure 28 LANDSAT 2 Plan of Snow Regions

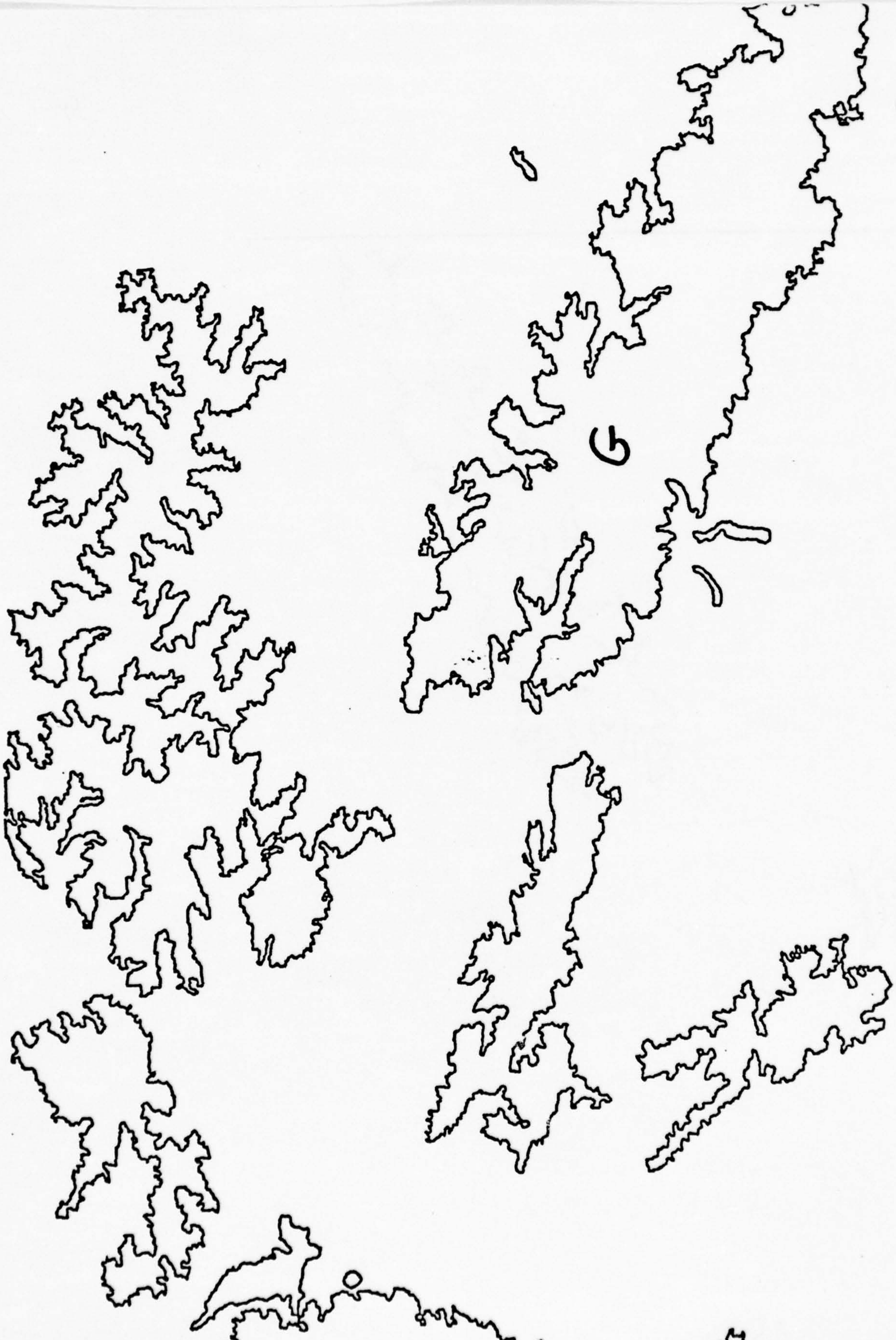


Figure 29 LANDSAT 1 Final Segmentation

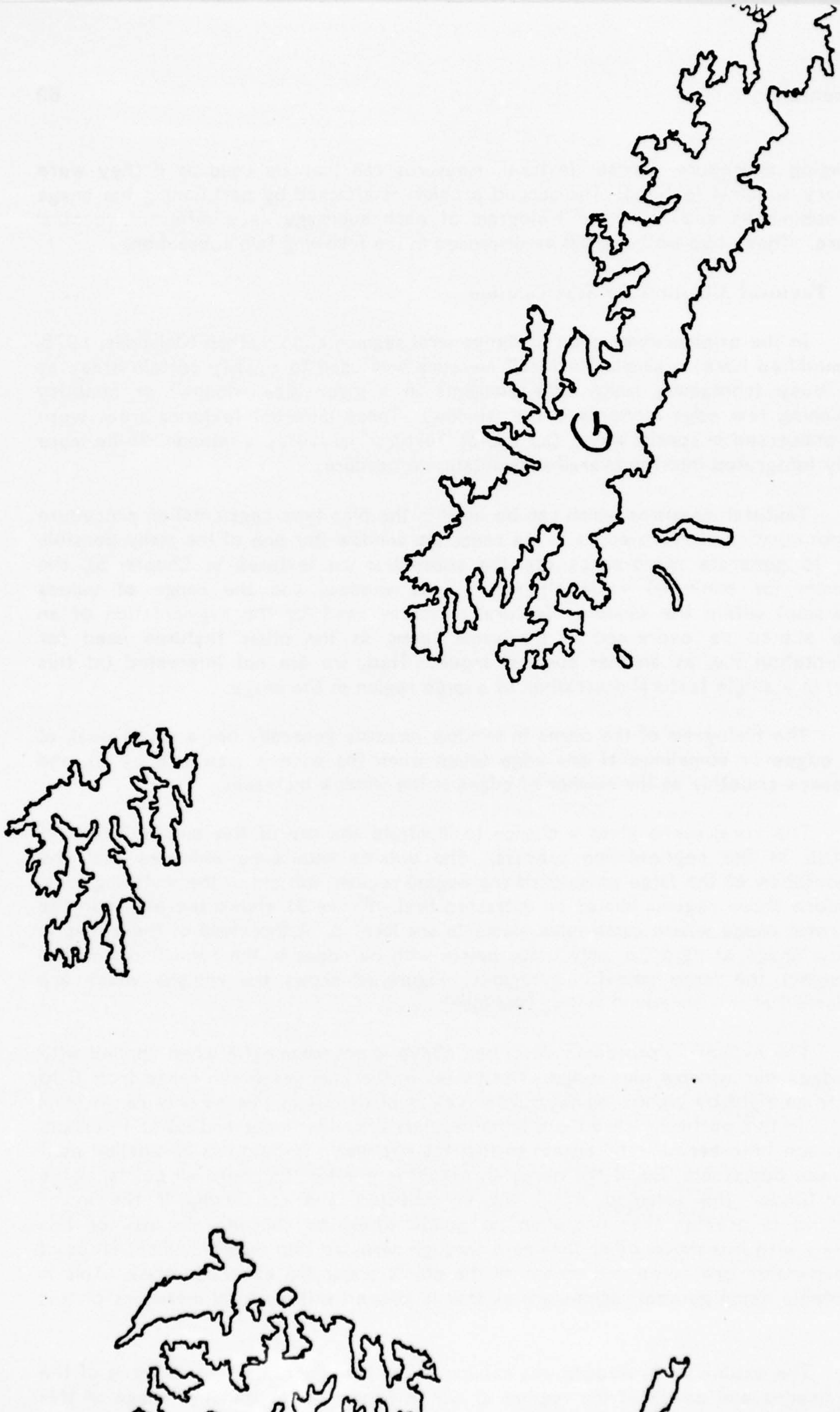


Figure 30 LANDSAT 2 Final Segmentation

averaging procedure. These "textural" measures can then be used as if they were ordinary spectral features. The second problem is attacked by partitioning the image into subimages and using the histogram of each subimage as a different spectral feature. These two methods will be discussed in the following two subsections.

4.5.1 Textural Measures for Segmentation

In the original description of the general segmentation method (Ohlander, 1975; and modified here) a simple "textural" measure was used to classify certain areas as very busy (containing many edge elements in a given size window) or nonbusy (containing few edge elements in the window). These different textured areas were then processed in special ways. Our use of "textural" measures is intended to be more tightly integrated into the overall segmentation procedure.

Textural measures which can be used in the plan type segmentation procedure include: number of micro-edges in the reduction window (for one of the many possible ways to generate micro-edges see the subsection on textures in Chapter 5), the maximum (or minimum) value attained in the window, and the range of values (excursion) within the window. Textural measures used for the segmentation of an image should be expressed in the same terms as the other features used for segmentation (i.e. as another spectral input). Thus, we are not interested (at this stage) in a single textural description of a large region of the image.

The histogram of the edges in window measure generally has a single peak at zero edges or sometimes at one edge (even when the window size is 8 by 8), and decreases smoothly as the number of edges in the window increases.

The rural scene gives a chance to illustrate the use of this simple "textural" measure in the segmentation process. The outside knowledge indicates that the segmentation of the large untextured (no edges) regions will aid in the matching task, therefore these regions should be extracted first. Figure 31 shows the points in the first rural image where micro-edge elements are located. A threshold of the edges in window image at zero (i.e. only those points with no edges in the reduction window) will select the large untextured regions. Figure 32 shows the regions which are segmented at this threshold setting (the plan).

The expansion procedure described above is not meaningful when applied with the edges per window plan image. The values in the plan image can range from 0 to 64 (for an eight by eight window), but the values of the original image only range from 0 to 1. In this particular case there is no problem since the upper and lower threshold values are both zero (i.e. no edges) so that the expansion method can be applied as it has been described. But if the upper threshold is greater than zero all points in the image (under the enlarged mask) will be selected, and conversely if the lower threshold is greater than one then no points would be chosen. Because of this problem with thresholds other than zero through zero, we skip the refinement steps of the expansion procedure and accept all the points under the expanded mask. This is acceptable since general regions are all that is desired with textural measures of this type.

The expansion procedure was applied to the plan generated for all three of the rural images and produced the regions shown in Figure 33 for the first image of this

Figure 31 Rural 1 Micro-edge Image

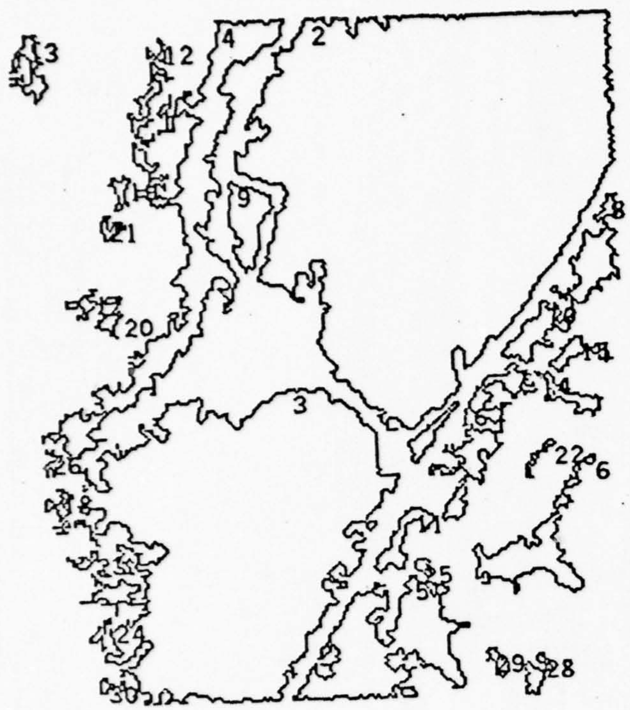


Figure 32 Plan for Smooth Regions in Rural 1 Image

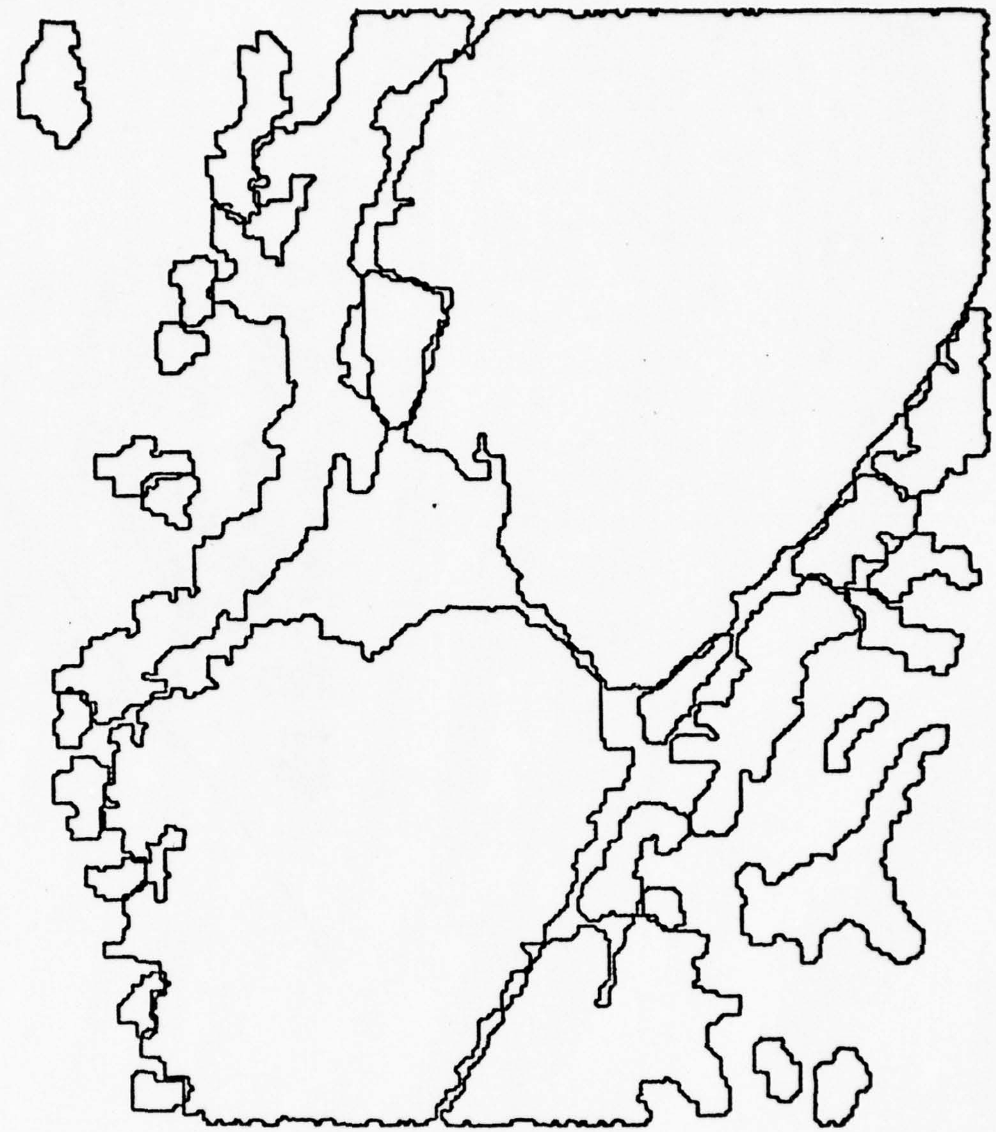


Figure 33 Segmented Smooth Regions in the Rural 1 Image

scene (the others will be presented later). The segmentation of the other regions which are needed for this scene will be discussed in the next subsection on partitioning of the image.

This smooth area extraction procedure is also applied to the SLR images and produces a partial segmentation; Figure 34 shows the first image and Figure 35 shows the second. These regions are sufficient for part of the task: find several key regions in both images for use in symbolic registration operations. But they are not enough for the second part of the task: the location of large texturally different regions. It seems that given enough simple "textural" operators the area in the upper right (dark, but with many bright regions, and high contrast) could be distinguished from the area on the left (many edges, but lower contrast, higher average intensity).

Another possible textural measure is the "excursion" measure (maximum in the window minus the minimum in the window). This measure should distinguish regions in the image with low contrast (low excursion values) from regions with high contrast (large excursion values). This measure was used in the SLR-1 image after the extraction of the nontextured (no edge elements) regions. The goal here is to separate the low contrast regions on the left side of the image (see Figure 3.11) from the higher contrast regions on the upper right side. But before we can perform this segmentation we must introduce another technique, partitioning, in the next subsection. Since this measure has no directly corresponding full size image, the expansion of regions in the plan is handled the same as the edges per window measure, i.e. there is no refinement step and all the points under the mask are accepted.

4.5.2 Segmentation with Partitioning

As the number of separate regions in an image increases, due to either decreasing the region size or increasing the image size, the amount of overlap of the peaks in the histogram associated with the separate regions increases. For example, in the urban images, the histogram of intensity does not exhibit a clear peak for the separation of the bright regions as seen in Figure 36. But there are clearly bright regions in the image (Figure 3.14). There are values (in the histogram) indicating bright values, but there is no separate peak for these bright regions. If we could decrease the number of separate regions included in the computation of the histogram, then a peak for the bright regions may become apparent. One way to reduce the number of different regions is to partition the image into subimages of smaller and smaller size until a desired peak appears or the histograms degenerate. This is the same technique used by Chow and Kaneko(1970) to select thresholds in medical images.

Figure 37 shows the histogram of the four quarters of the images. The histogram labeled "1" is the top left quadrant of the image, "2" is the top right, etc. There is still no separate peak in any of the four quadrants. The peak from 220 to 248 in the top left histogram does not meet the Ohlander criteria for an acceptable peak (peak maximum:peak minimum must be 2:1 or greater), but could be acceptable with less constrained peak criteria.

Figure 38 is the set of histograms when the image is divided into 9 subimages. In these, the histogram labeled "1" is the top left ninth of the image, "3" is the top right, "5" the center, etc. In this set of histograms, there are four acceptable bright

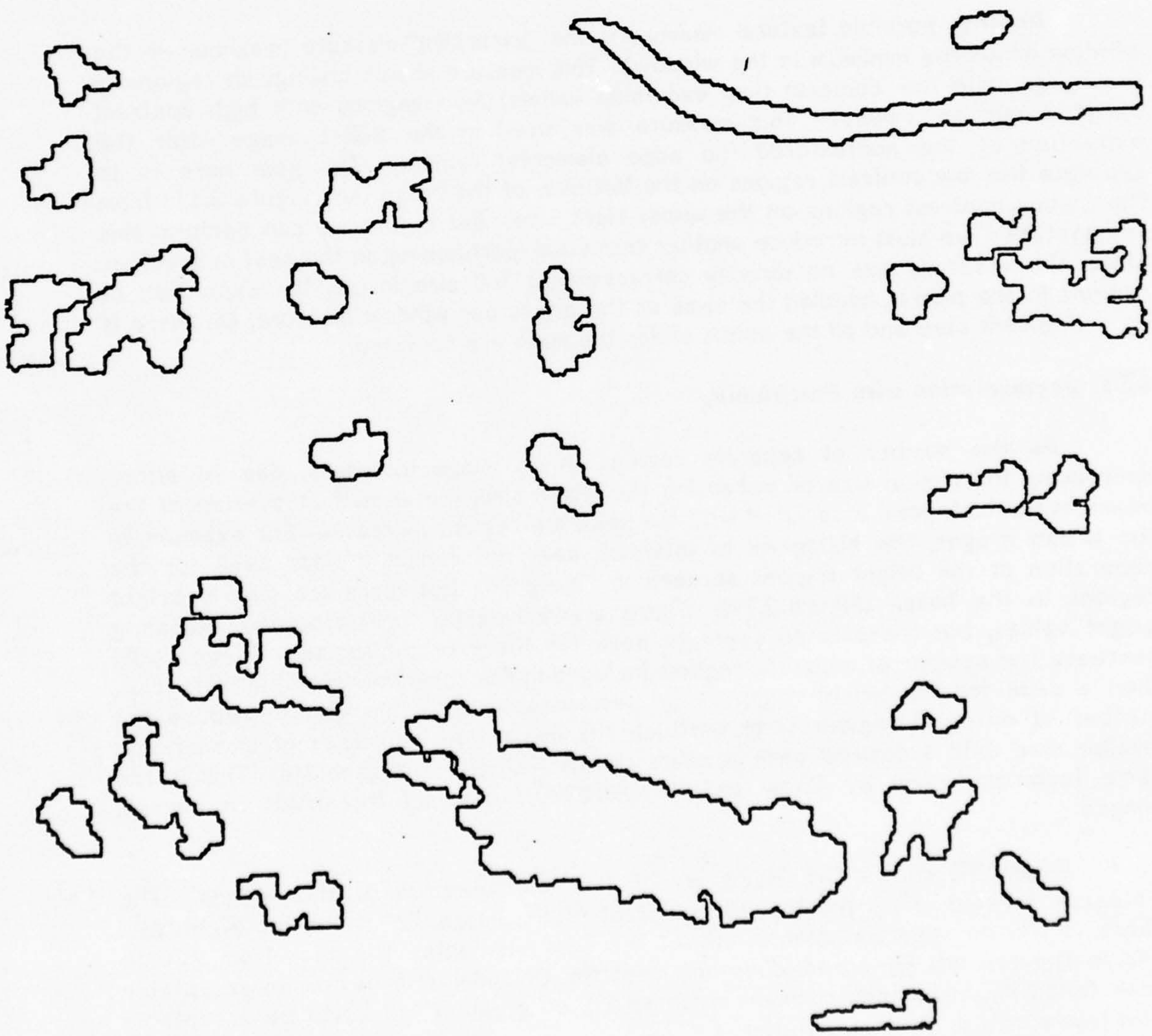


Figure 34 SLR 1 Segmented Smooth Regions

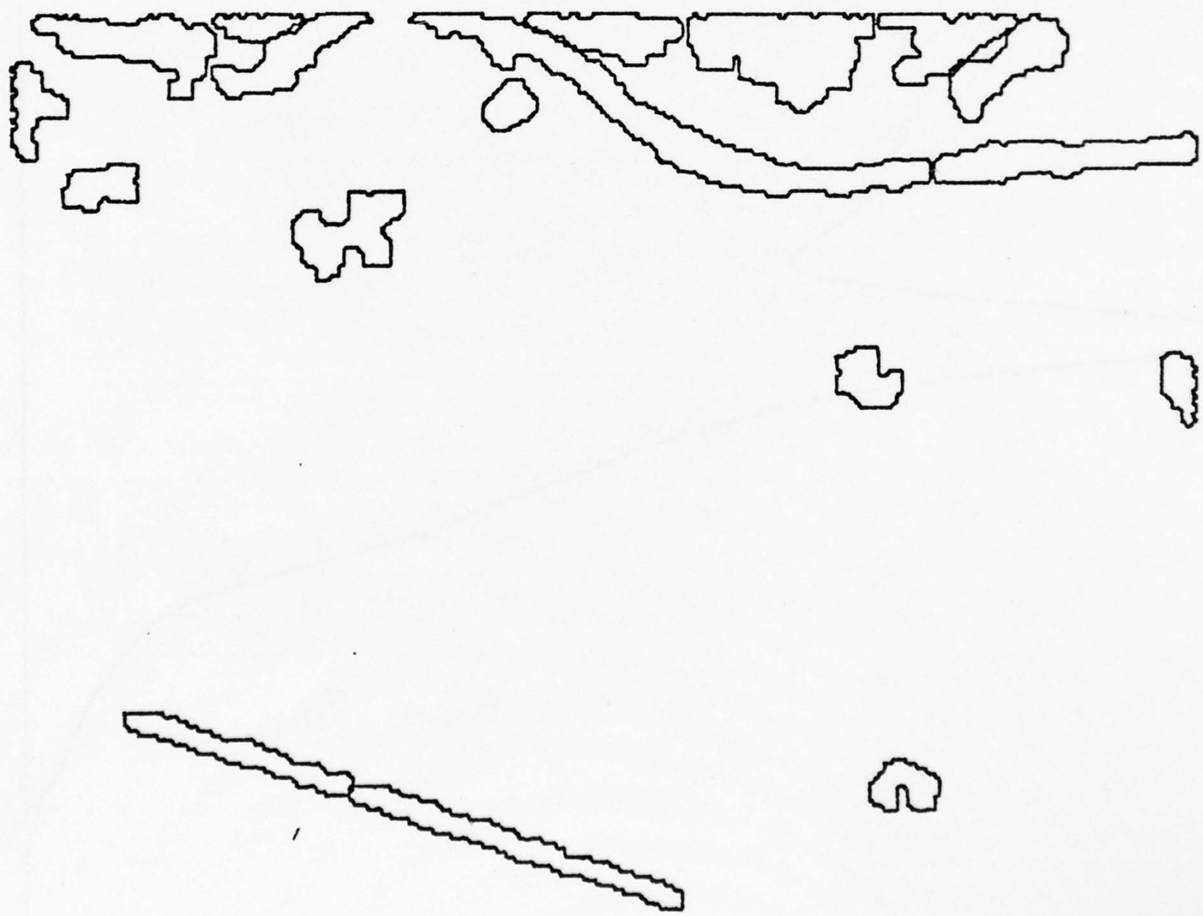


Figure 35 SLR 2 Segmented Smooth Regions

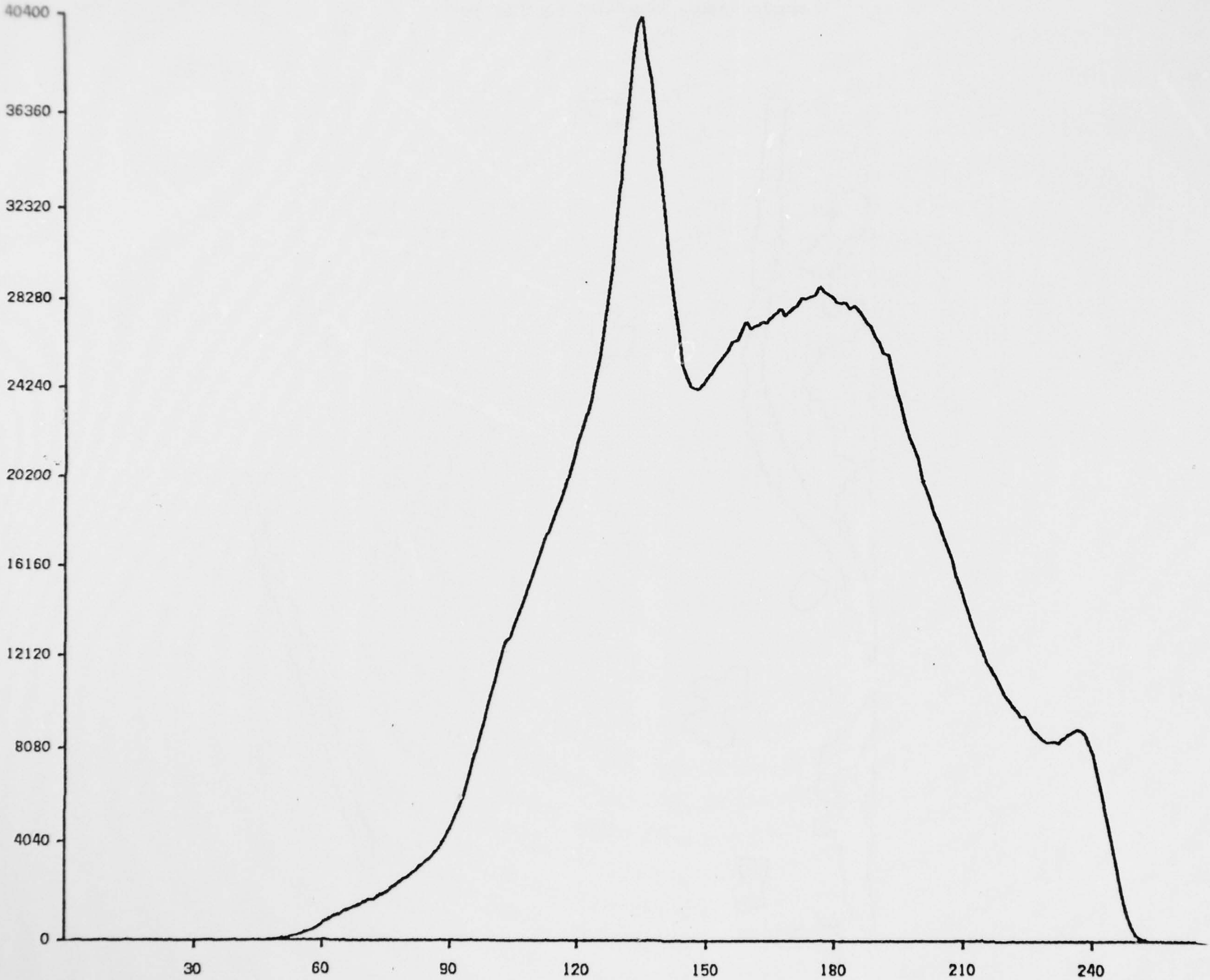


Figure 36 Histogram of Urban Reduced Image Intensity Only

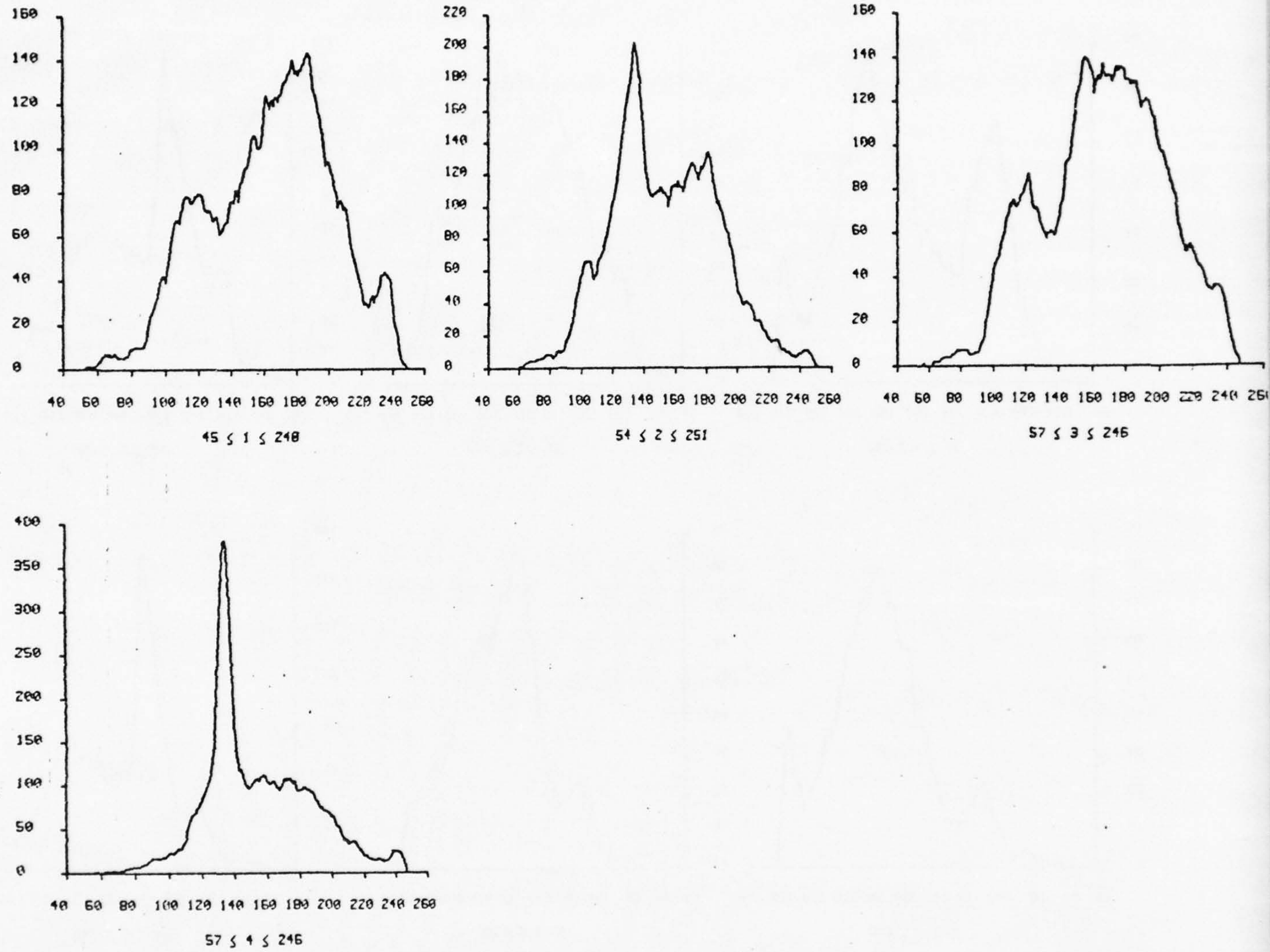


Figure 37 Histogram of Urban Intensity Image Partitioned into Four Subimages

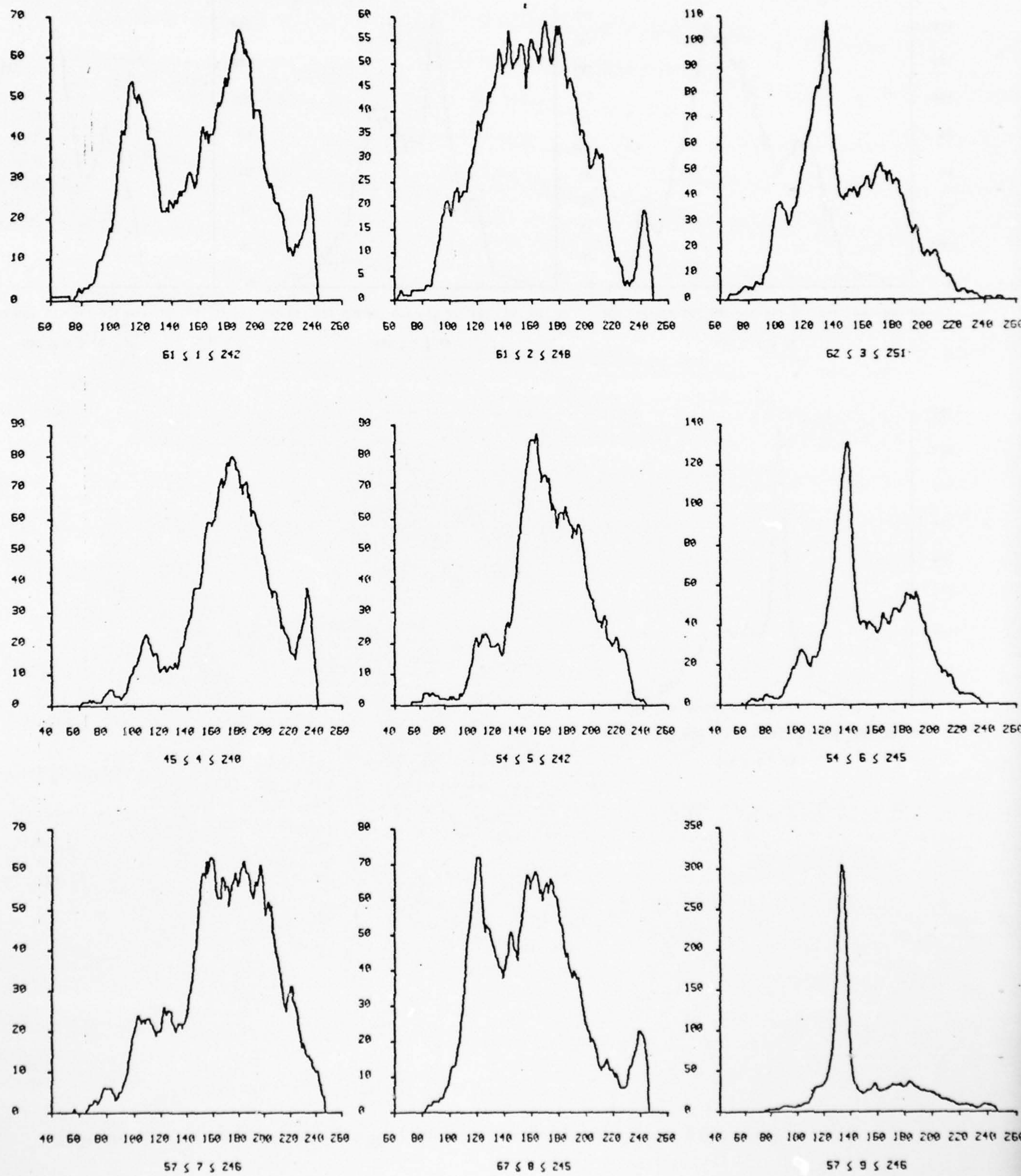


Figure 38 Histogram of Urban Intensity Image Partitioned into Nine Subimages

peaks (in "1", "2", "4", and "8"). Since the selected threshold values will be used over the entire image, a threshold range of 220 to 256 was selected to cover all four of these bright peaks. There are other peaks in these histograms, but, since were are looking for a way to separate only the bright regions, the others are ignored.

This threshold value (220-256) is applied to the entire urban-2 image and produces the regions shown in Figure 39 for the plan and Figure 40 for the full size image. This selects the group of round bright regions (top center) and the long thin region (bottom center) which are desired as anchor regions in the task for the urban scene.

After the initial extraction of smooth regions, the segmentation task for the rural images is the same as the urban images: find the bright regions. But there are a few differences; the required regions are much smaller and bright regions have low values rather than high values. After the smooth regions are extracted, the histogram for the remaining scene (Figure 41) shows no peak of bright points. This is the histogram of the full size image rather than the planning image since the desired regions are as small as 250 pixels, which would be be too small in the plan image to be considered useful (i.e. three or four pixels).

There are still no separate peaks (for the bright regions) in the two by two partitioning Figure 42 and the three by three partitioning Figure 43 of the image. But an analysis of the histograms in Figure 42 shows that two of them, "2", and "4", corresponding to the right side of the image, indicate that bright points occur in those quadrants, while the other two ("1", and "3") show that fewer bright points occur in the two left quadrants. If the histogram for the lower right quadrant ("4") is thought of as the sum of the histograms of the individual regions (each with a single peak), then we can assume that the points below about 25 or 30 come from the bright regions, and there is probably another peak centered around 30 covering the less bright regions (or points partially in a bright region). Also there is the large background peak which appears in the other quadrants ("1", and "3").

If we use these assumptions we can select a threshold of about zero to 25 to extract the bright regions. This process is an ad hoc method for the extraction of specific peaks and can not be considered for the extraction of general segmentation peaks. But when the segmentation procedure is directed to find the brightest (or darkest) regions such ad hoc techniques can be used. If the partitioning were carried to the extreme, then the division between the bright regions and the background would become apparent. This would occur when the partition included only the bright region and the background or only the bright region alone. Figure 44 shows the bright regions which were extracted with this threshold.

We can now return to the SLR-1 segmentation using the excursion planning image. The three by three partition histograms for the remaining image are given in Figure 45. In most of the sections there is a large peak centered near 20 (the low contrast regions) and a much smaller peak near 55, except the top right ninth (labeled "3") has a large peak around 60. If we then segment the image twice using each of the peaks in this ninth, then we can extract the large high contrast areas and low contrast areas. Figure 46 shows these two types of regions.

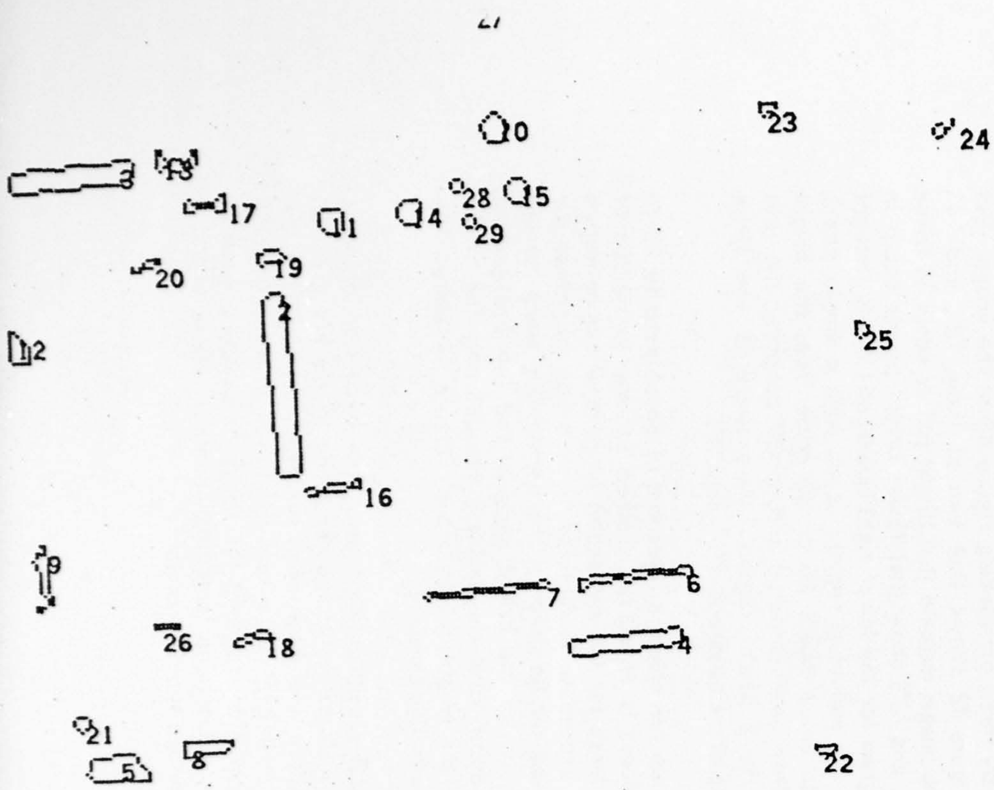


Figure 39 Urban 2 Plan for Segmentation of Bright Regions

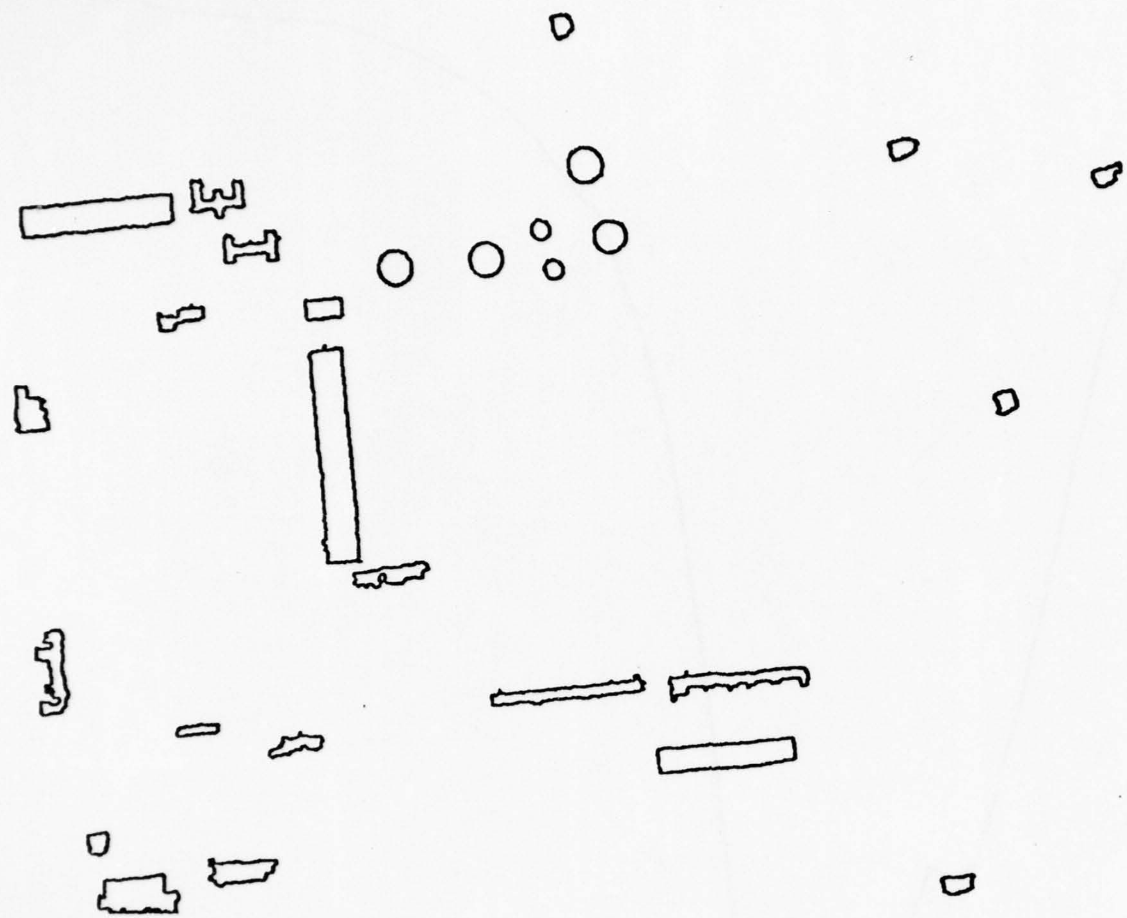


Figure 40 Urban 2 Bright Regions

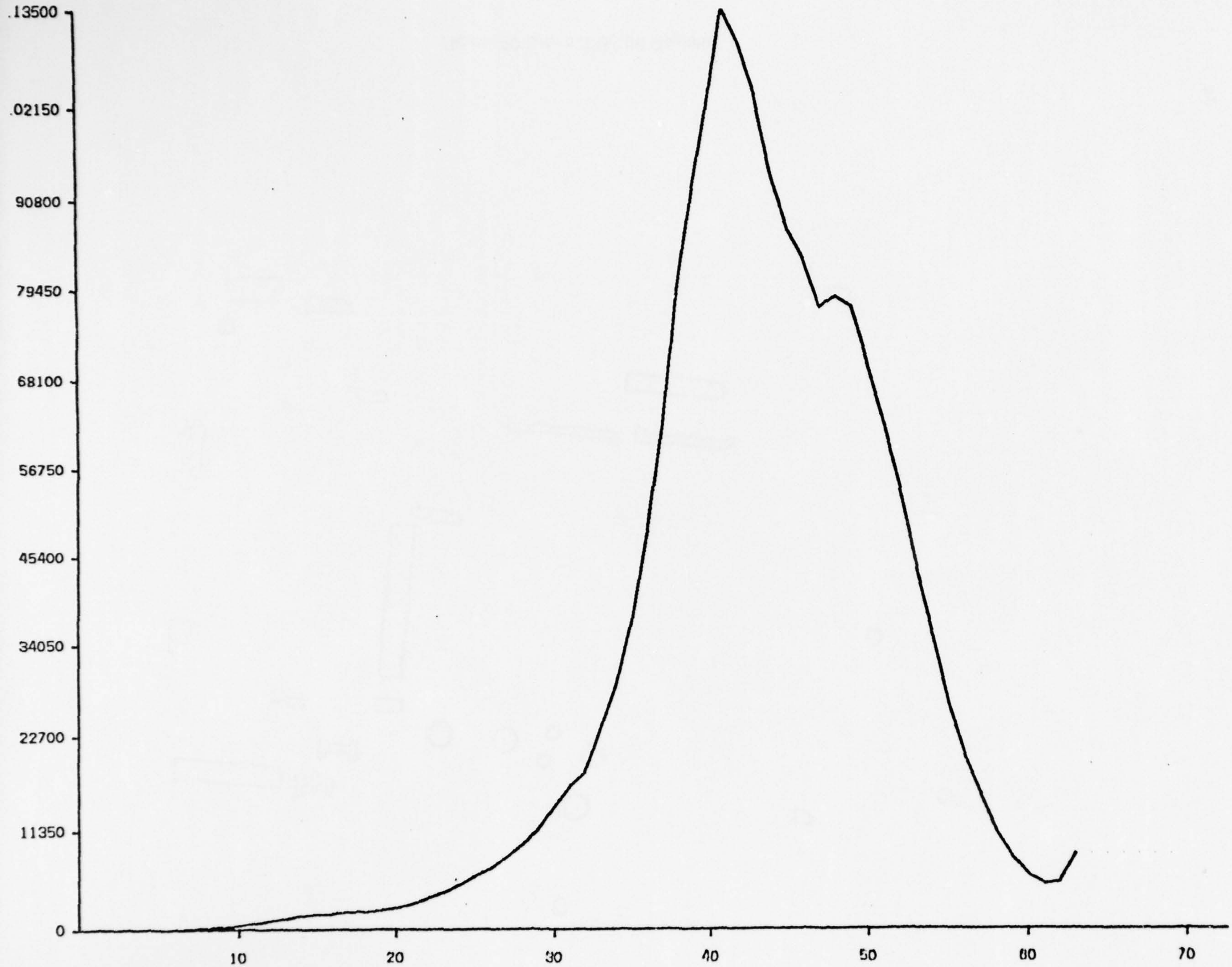


Figure 41 Rural Image Histogram of Non-Smooth Portion of Image

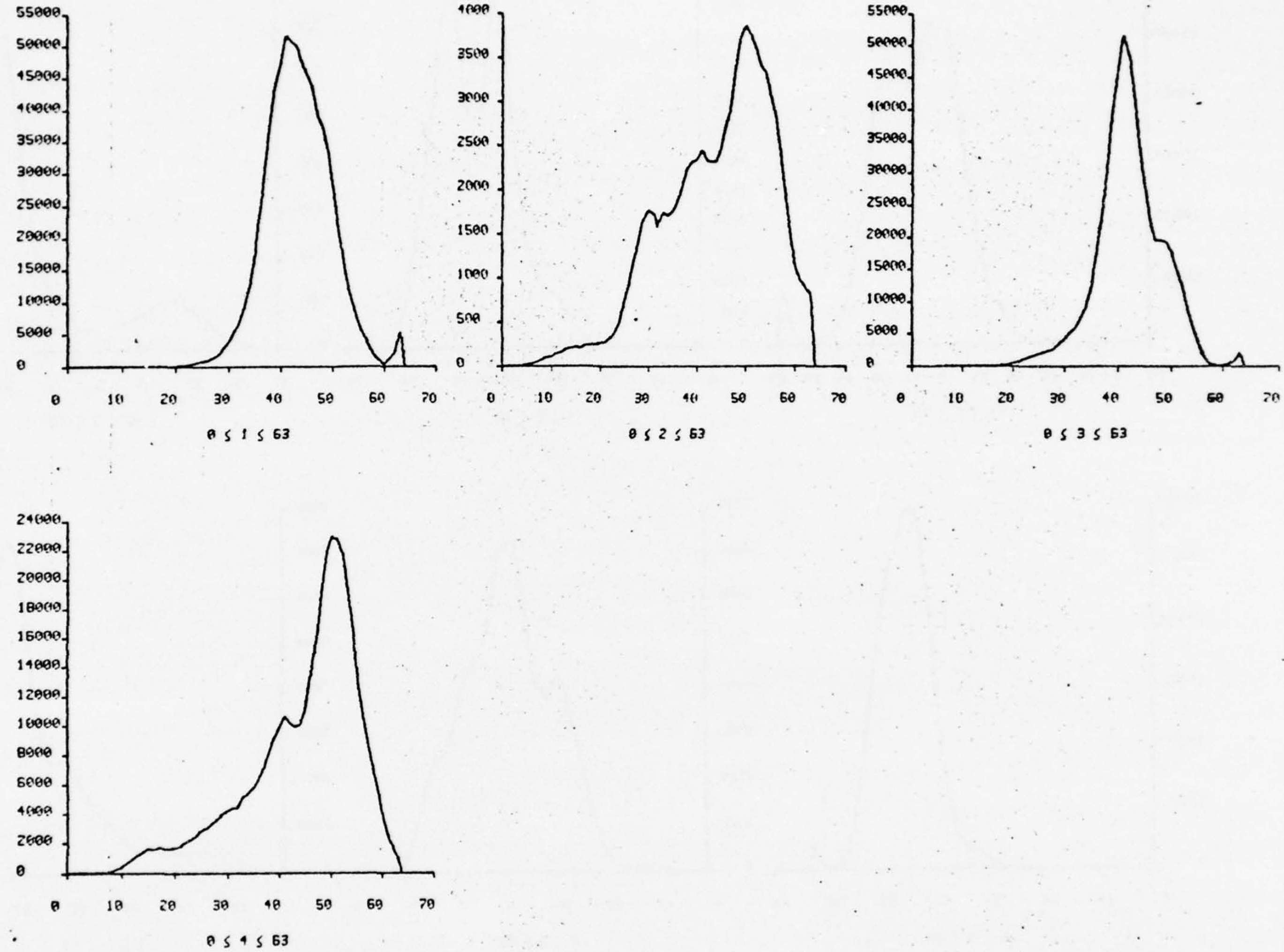


Figure 42 Rural Image Histogram of Four Subimages

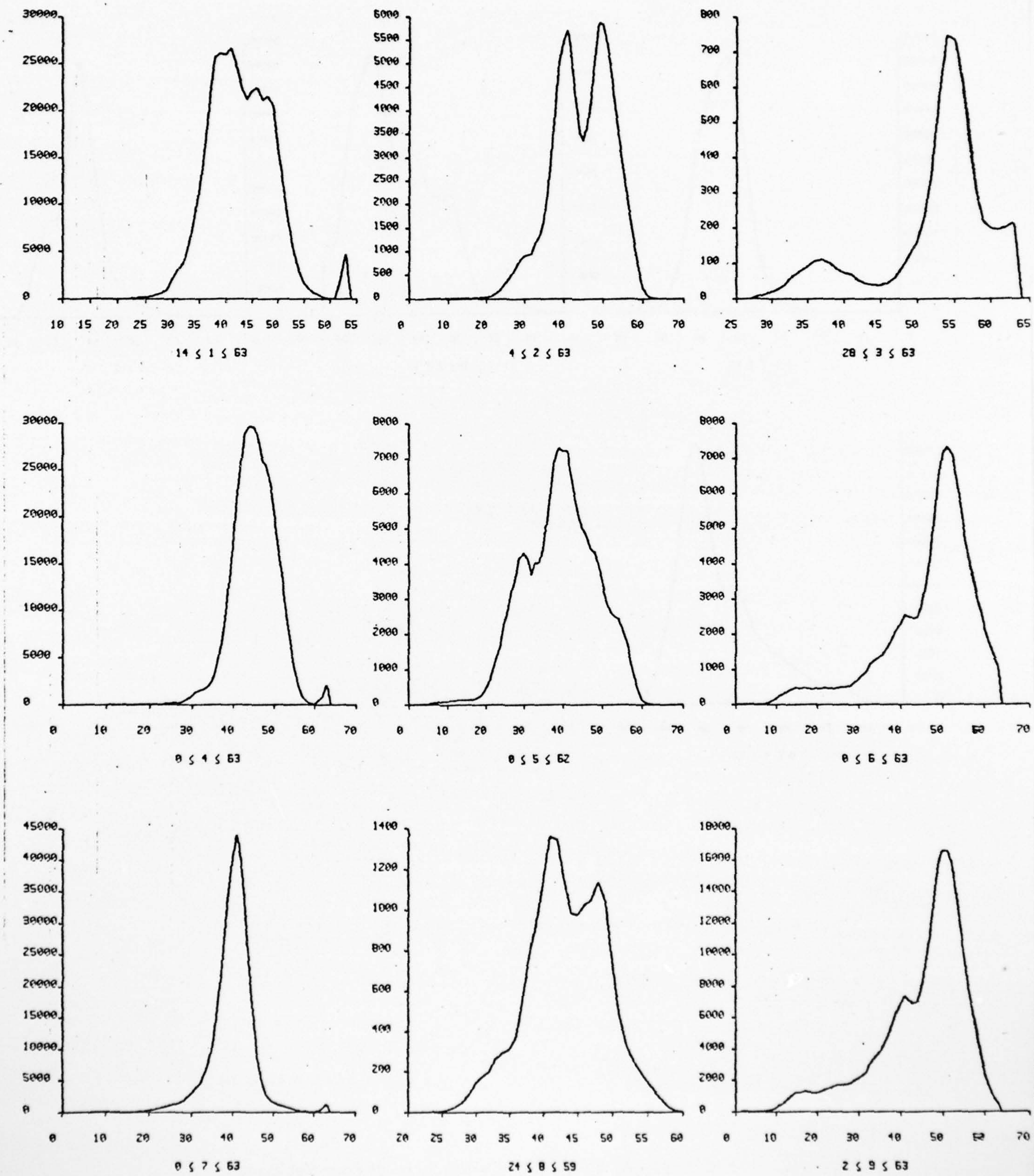


Figure 43 Rural Image Histogram of Nine Subimages

Figure 44 Rural 1 Bright Regions

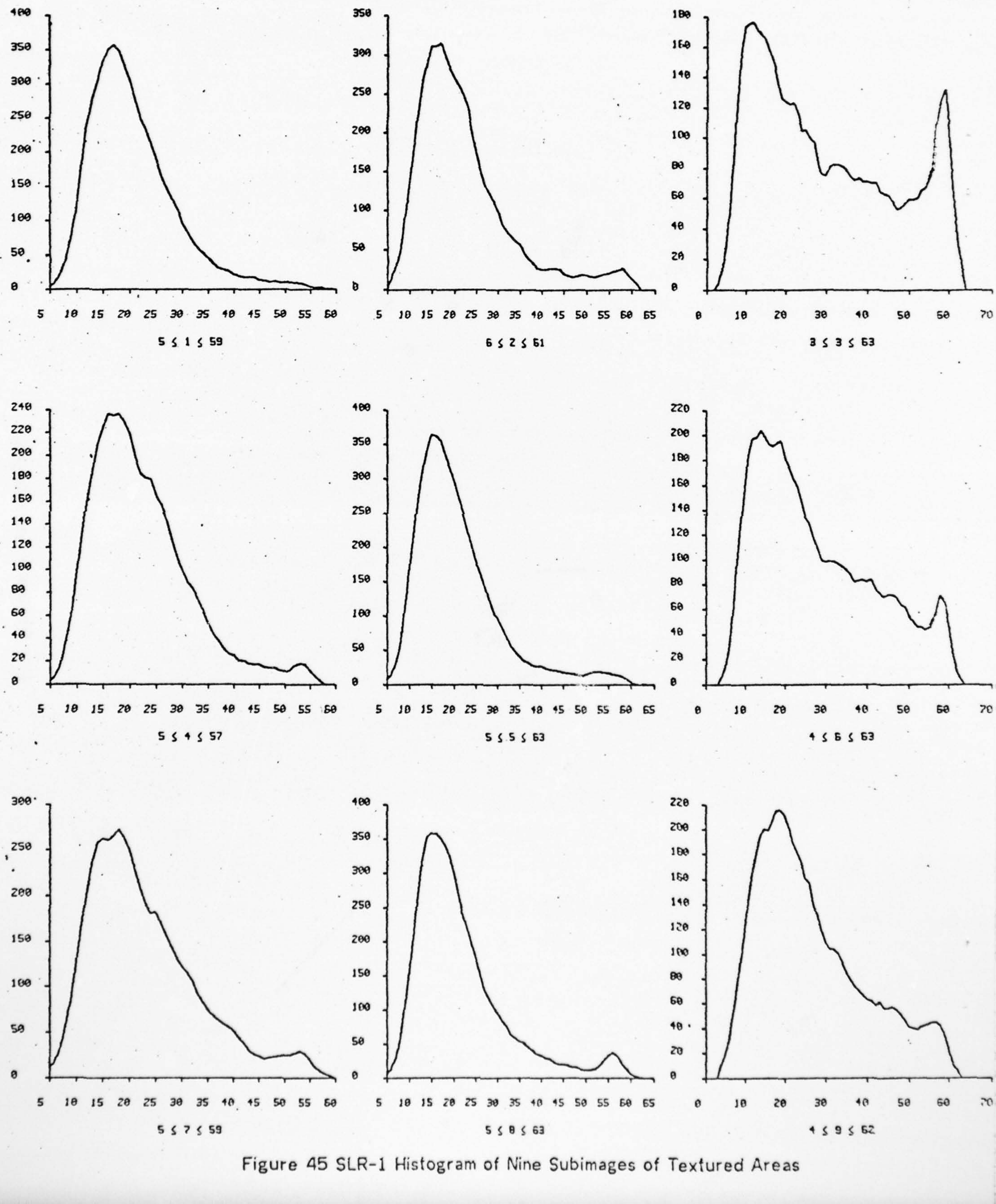


Figure 45 SLR-1 Histogram of Nine Subimages of Textured Areas

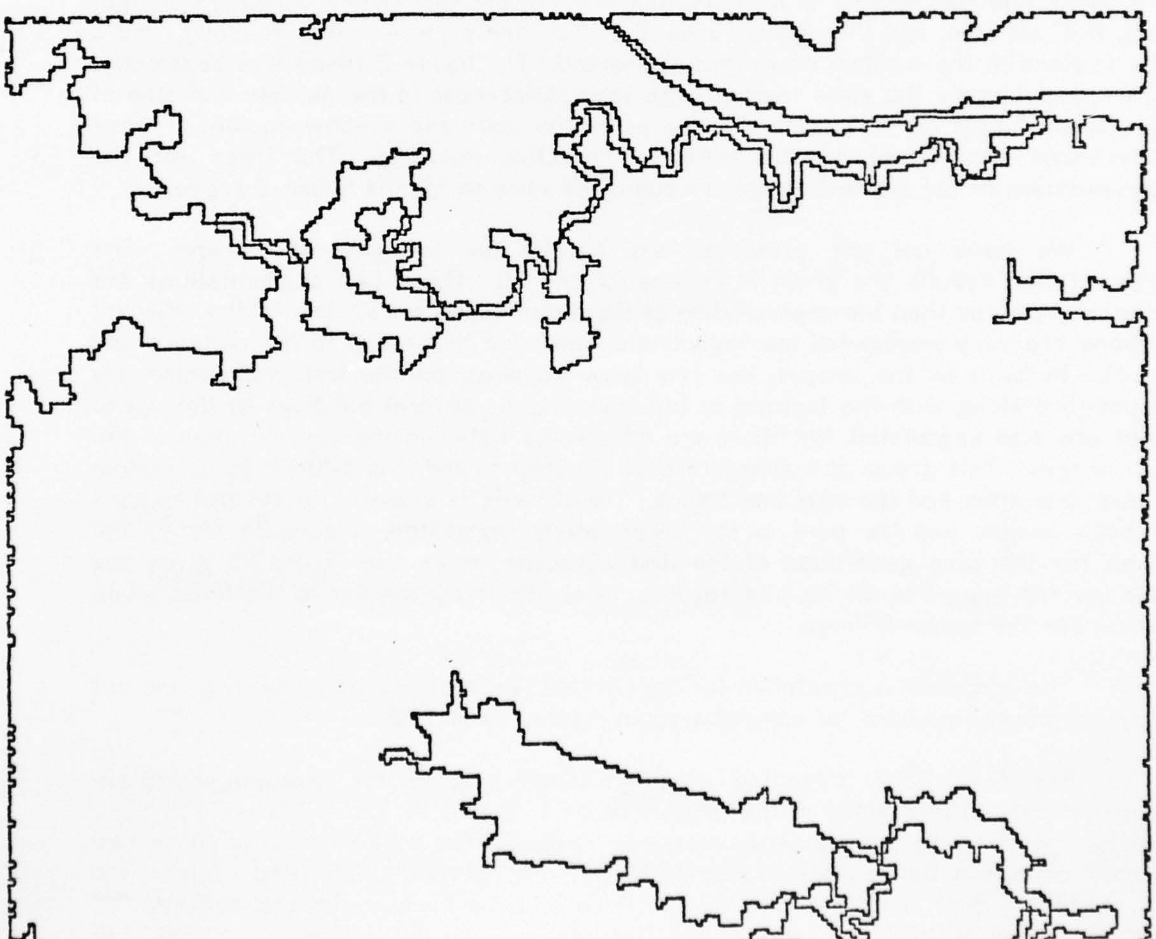


Figure 46 SLR-1 Regions Segmented Using Excursion Parameter

4.6 Results

This final section on segmentation will present the segmentation results for the thirteen images which were presented in Chapter 3 plus the segmentation results for the "pier" area of the two urban images. Some of the segmentations have already been presented, but will be given again in this section (without discussion) for completeness.

The segmentation for the house-2 image has already been presented along with the times for all the operations. Figures 47 and 48 give the final segmentation for the two house images. The segmentation of the house-1 image produced the large sky, lawn, and roof areas. In addition, four wall regions (above the window, the right side, the left side, and the middle), several bushes, the chimney, door, shadows, and a few regions in the window area were segmented. The house-2 image was segmented into approximately the same regions, with some differences in the number and size of the "bushy" regions, and some differences in the door and window regions. These differences should not be too much for effective matching. The times for the segmentation of the house-1 image are about the same as for the house-2 scene.

We have not yet presented any results for the cityscape scene. The segmentation results are given in Figures 49 and 50. These two segmentations are generally poorer than the segmentation of the house scene since many of the adjacent regions are very similiar (all the regions are somewhat bluish due to the distance and haze). In both of the images, the two large buildings on the left and center are segmented along with the building in the lower right. Several buildings in the upper right are also segmented, but there are differences between the segmentation in the two images. This group of buildings are all silver-gray and it is difficult to determine where one ends and the next one begins. The hill side is broken into several regions in both images and the park in the lower left is segmented. Figure 58 shows the times for the plan generation of the first cityscape image and Figure 59 gives the time for the expansion of the plan regions. These times are similiar to the times given earlier for the house-2 image.

The complete segmentation for the LANDSAT scene has already been given, but they are presented here for completeness in Figures 51 and 52.

The SLR scene presented a more difficult problem for accurate, complete segmentation. The regions in the original image (Figure 3.11 and 3.12) are not very well defined (except the dark and untextured regions). The segmentation of these two images produced the regions in Figures 53 and 54. Several untextured regions are segmented in both images, especially the "river" in the lower right, the reverse "C" shaped region in the lower left, and the "runway" area on the left (in two pieces). In the first image the "runway" regions include the surrounding areas which are not included in the second image. The larger differently textured regions are rather general and amorphous, but that is the same way that they appear in the original image.

The segmentation of the first rural image has already been given in two parts. The complete segmentation of all three are given in Figures 55, 56, and 57. The

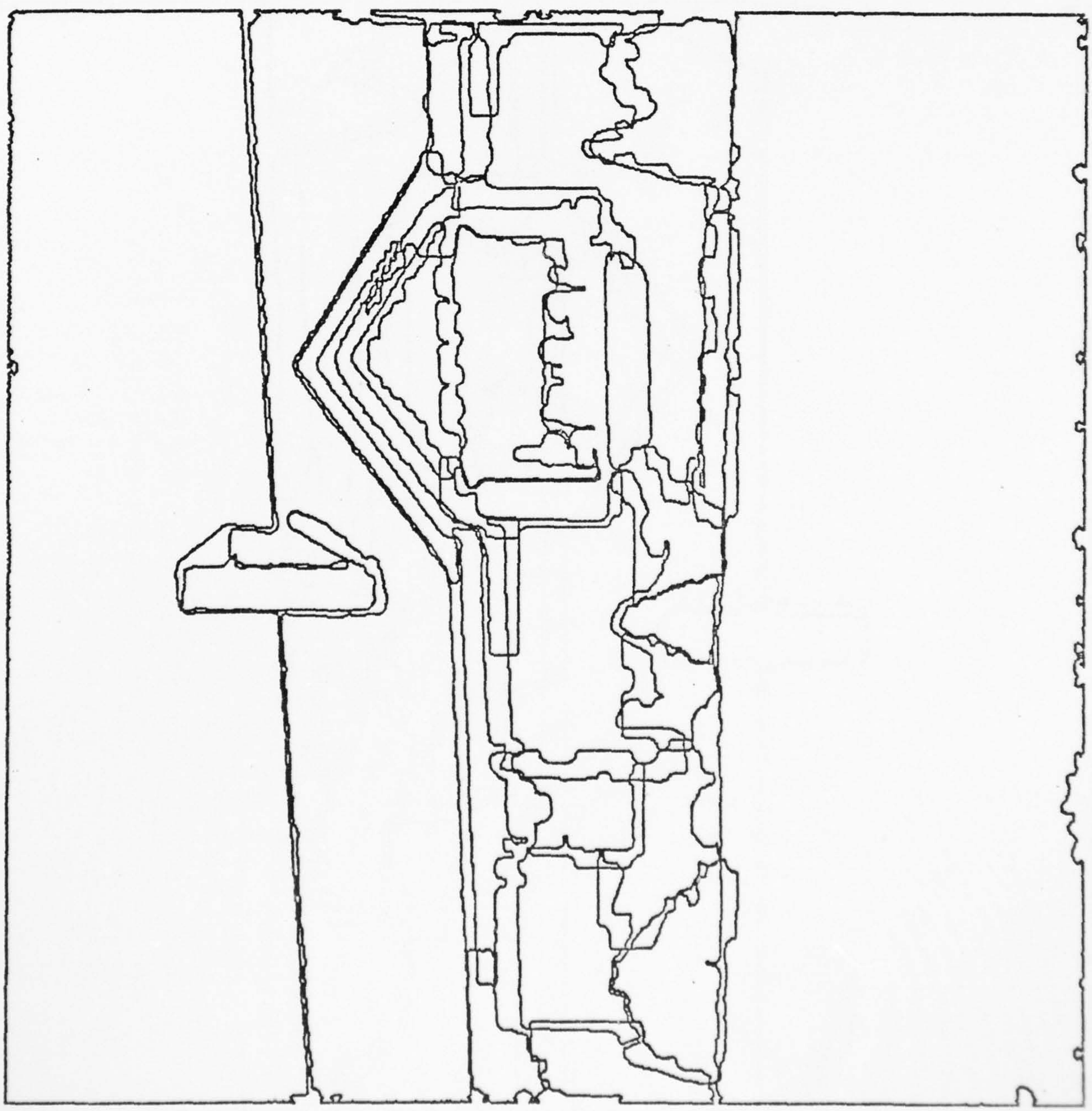
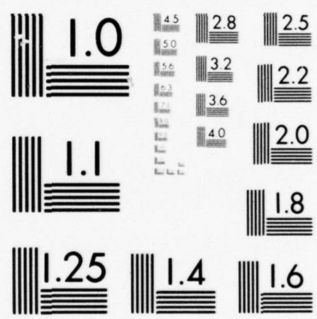


Figure 47 House 1 Segmentation

AD-A038 843 CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER --ETC F/G 15/4
CHANGE DETECTION AND ANALYSIS IN MULTI-SPECTRAL IMAGES,(U)
DEC 77 K E PRICE F44620-73-C-0074
UNCLASSIFIED AFOSR-TR-77-0448 NL

2 OF 3
AD
A038843



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

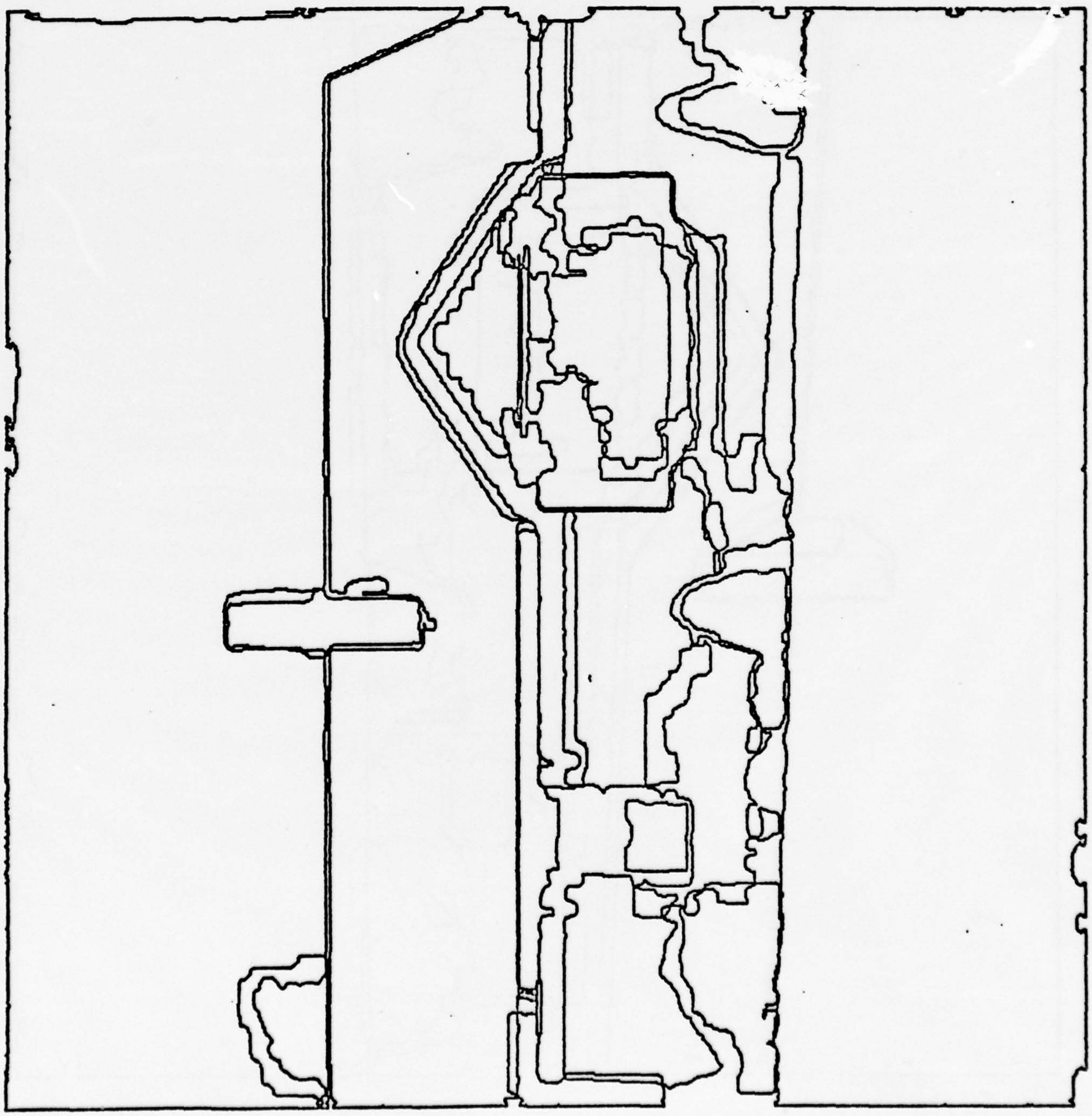


Figure 48 House 2 Segmentation

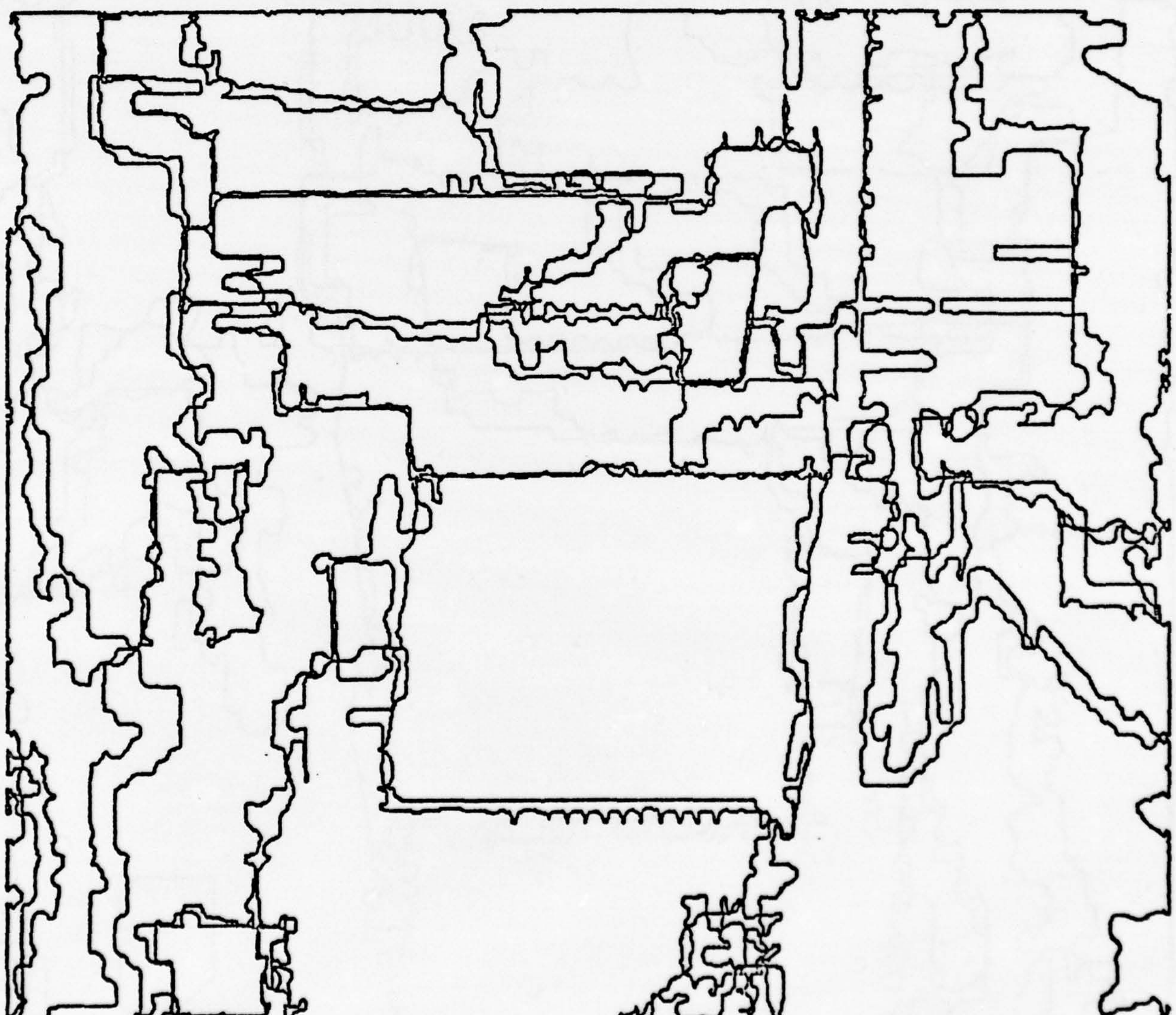


Figure 49 Cityscape 1 Segmentation

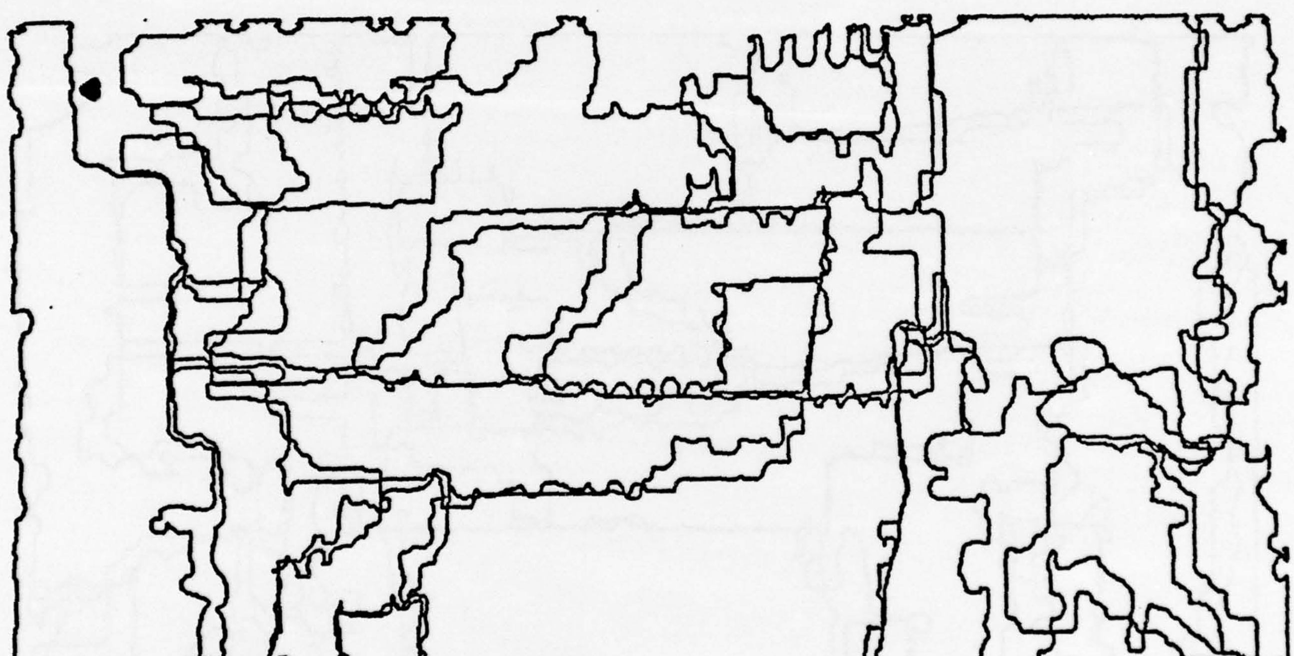


Figure 50 Cityscape 2 Segmentation

Operation	Millions of Operations	Percent of Total	Number of Times Used
Histogram Computation			
Generation of array	12.32	29.2	135
Smooth array	10.26	23.4	135
Other	0.33	.8	15
Peak Selection	7.88	18.7	15
Threshold	1.55	3.7	15
Smooth	3.51	8.3	15
Region Selection			
Initialize	3.86	9.2	15
Select a region	0.92	2.2	31
Save masks	1.50	3.6	43
Total	42.13	--	

Figure 58 Timing Summary for Plan Generation Cityscape 1

Operation	Millions of Operations	Percent of Total	Number of Times Used
Enlarge Small Mask	1.62	.7	30
Large Mask Threshold	32.32	13.8	30
Smooth Large Mask	151.18	64.6	90
Extract Regions	49.02	20.9	30
Total	234.14		

Figure 59 Expansion Timing Cityscape 1

method of segmentation of all three images was the same as for the first image; extract the untextured regions with the plan, then extract the bright regions from the full size image. Many regions are extracted in all three images. The two large regions on the top and bottom of the scene, the "river" on the left of the large regions, and several of the bright regions in the center are found in all three. Many of the houses in the lower right are segmented in the second and third image, but only a few are extracted from the first image. There are some differences in which smaller untextured regions are segmented in the three images, but most of the larger regions are the same. The lower left of the first image is more textured than that area of the other two, so that

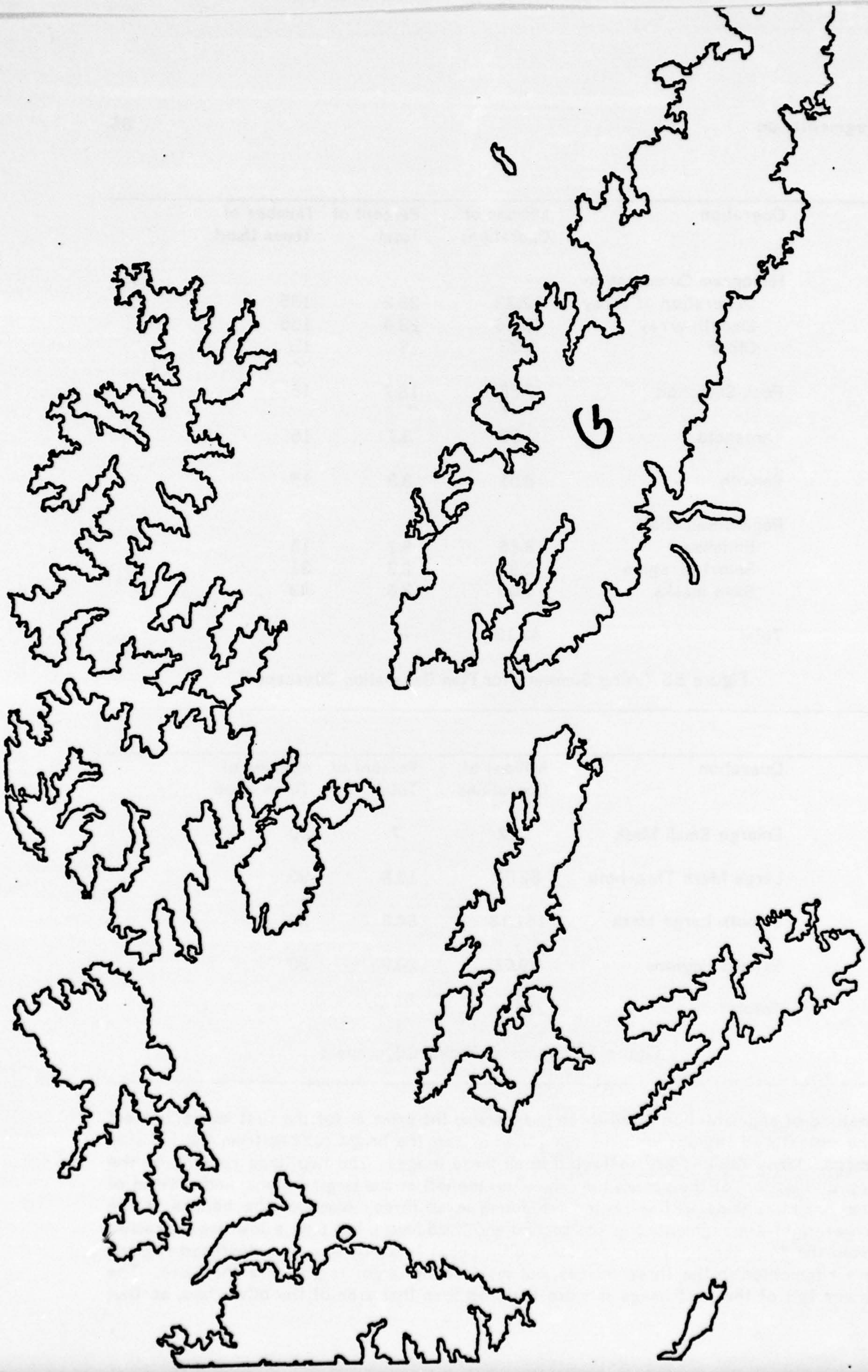


Figure 51 LANDSAT 1 Segmentation

Figure 52 LANDSAT 2 Segmentation

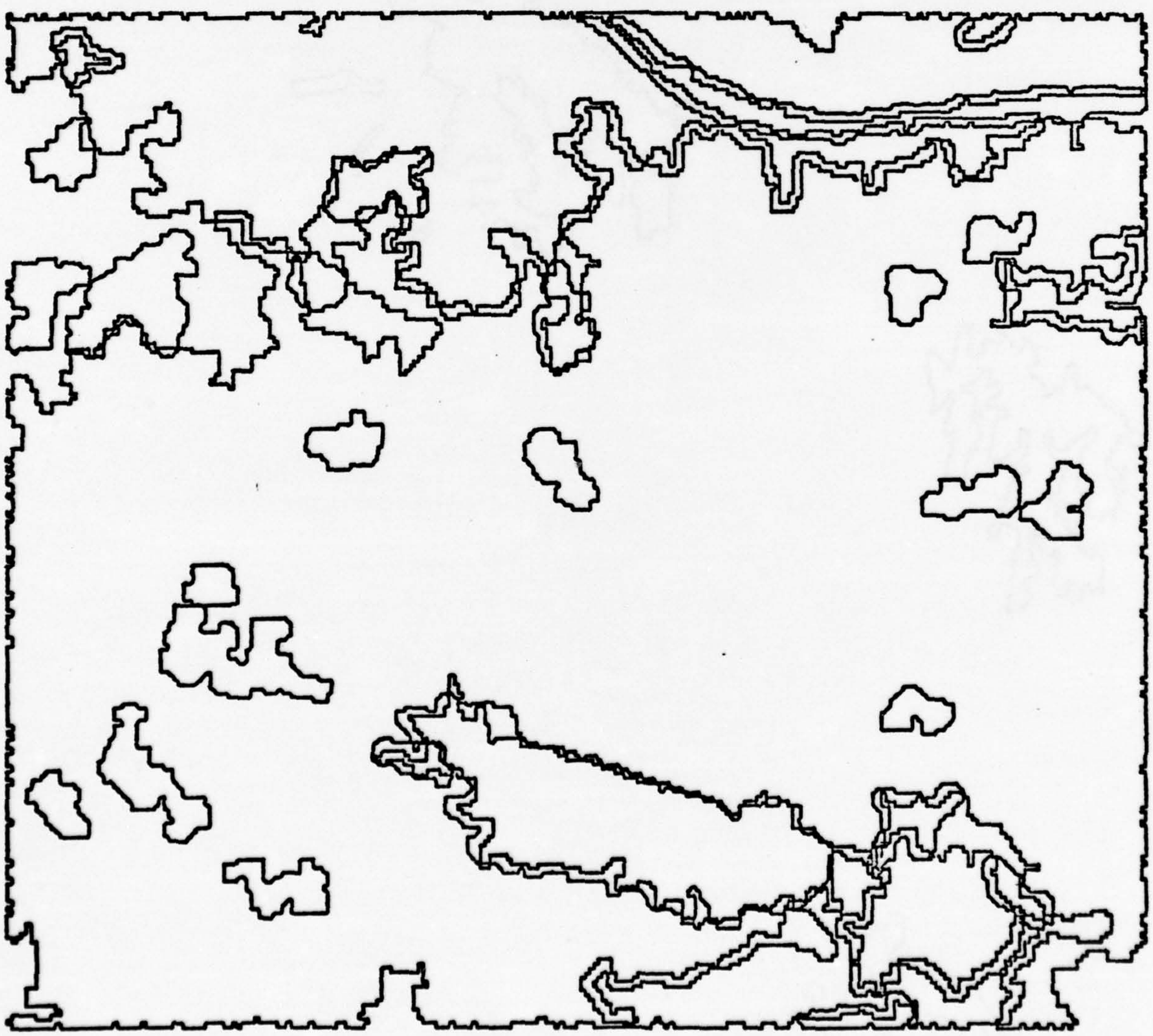


Figure 53 SLR 1 Segmentation

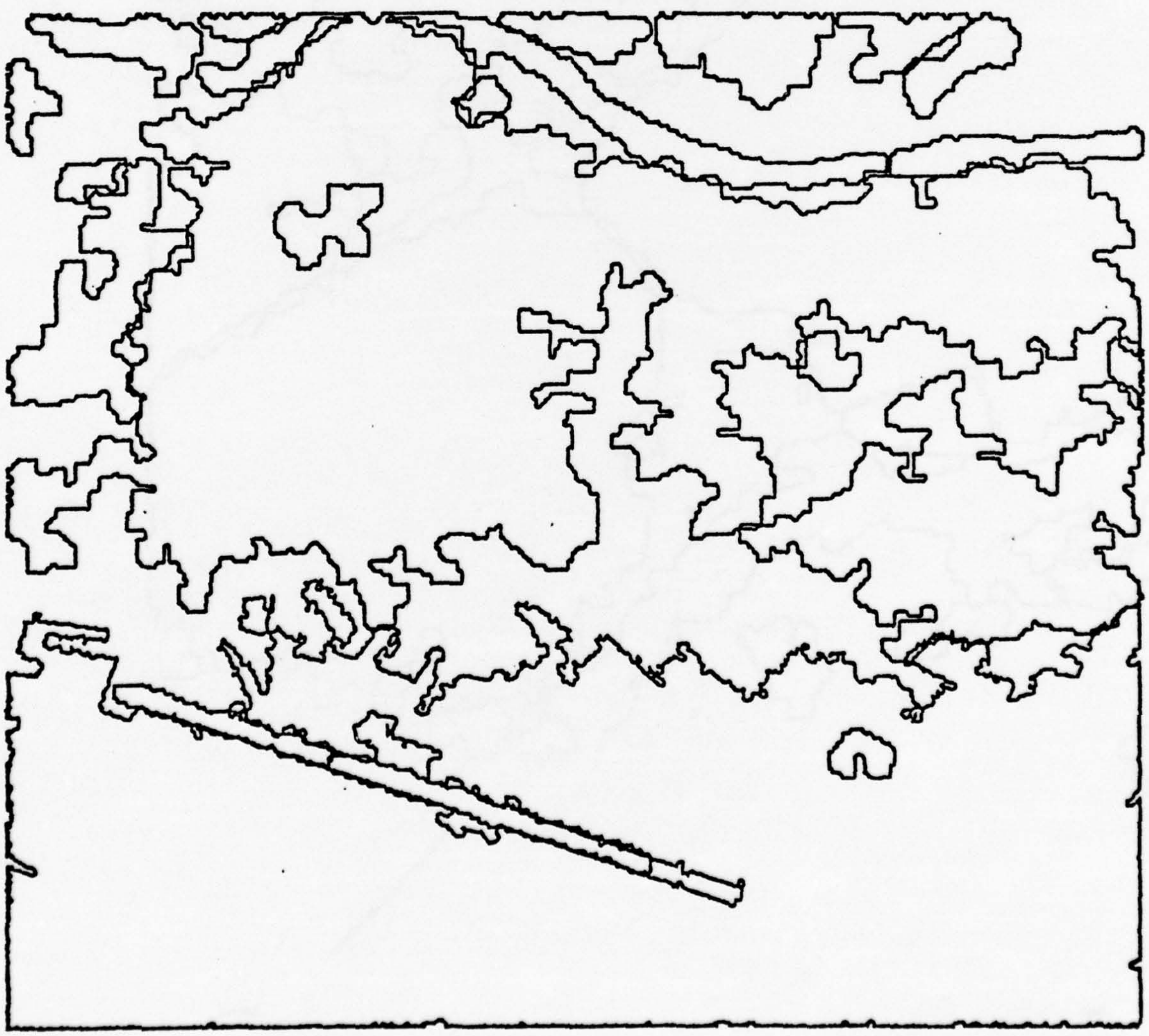


Figure 54 SLR 2 Segmentation

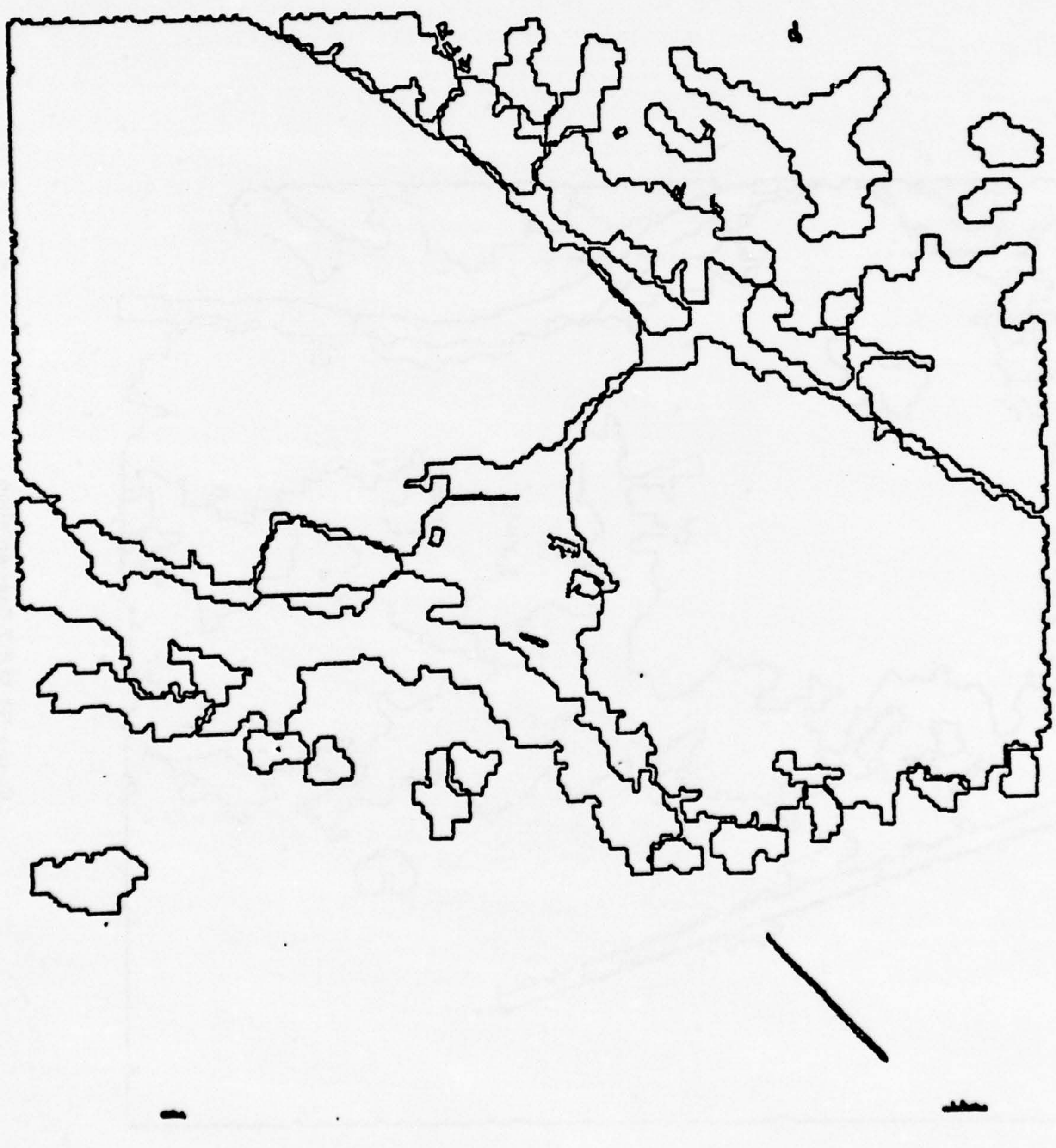


Figure 55 Rural 1 Segmentation

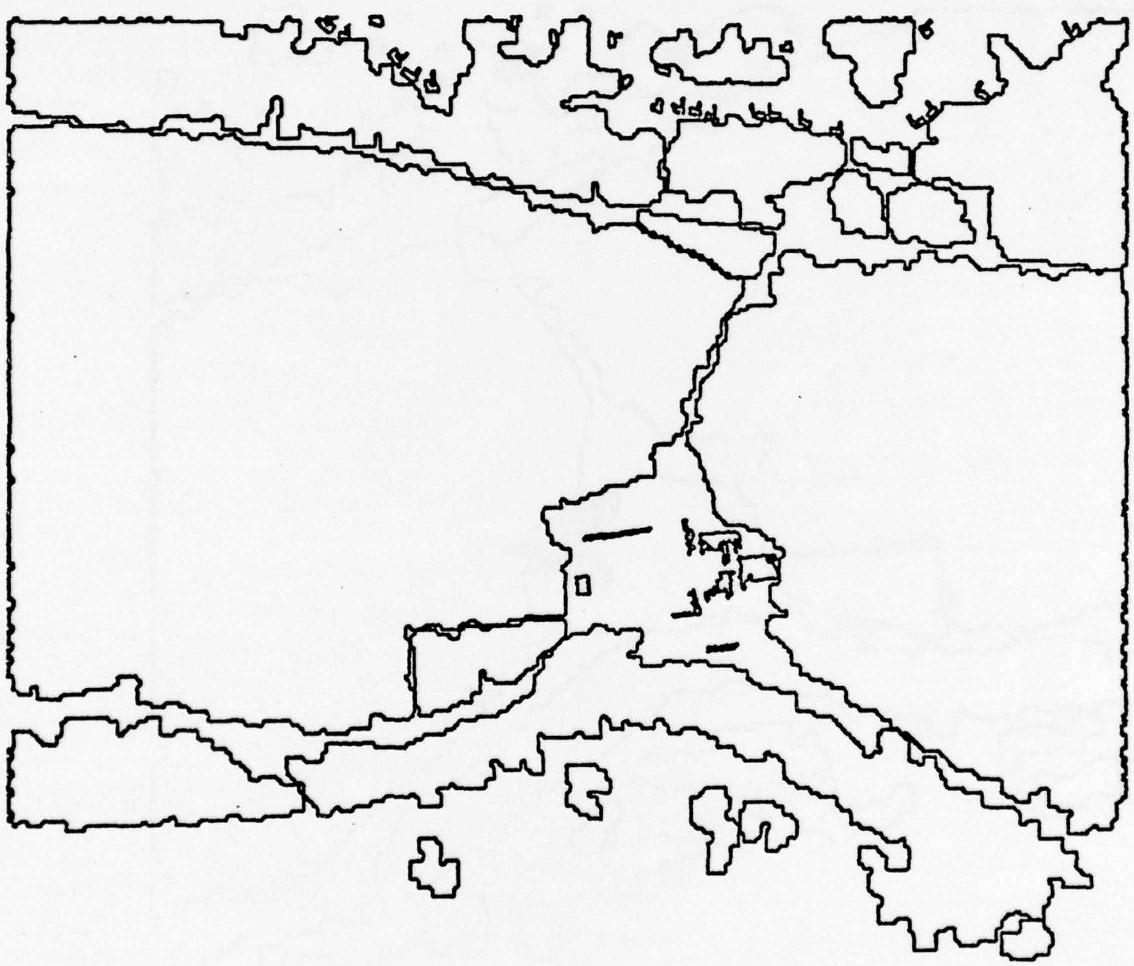


Figure 56 Rural 2 Segmentation

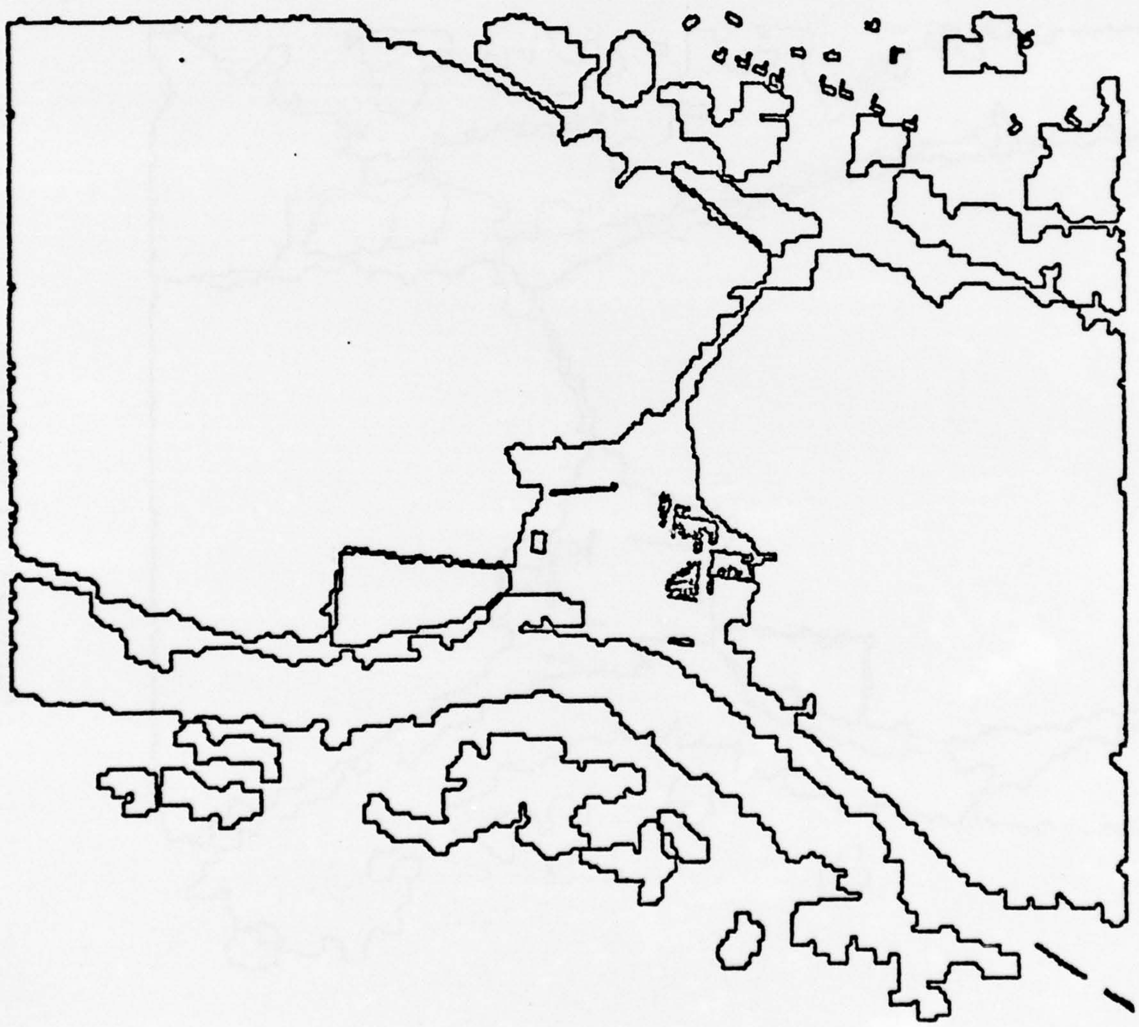


Figure 57 Rural 3 Segmentation

the large region on the bottom and the "river" region do not extend into this area. Also, the "river" is broken into two parts in the second image.

The final scene is the urban scene. The segmentation processing on the entire image is limited to the extraction of a few bright regions for use in limiting the area to be used in future analysis. Figures 60 and 61 show the bright regions extracted in the two urban images. The group of round, bright objects are extracted along with a large rectangular region in the lower right, and several other bright regions (buildings). Some of the round regions in the first image are not completely extracted because of the shadows due to the lower sun angle. Later matching procedures will use some of these regions to limit the area to be analyzed as the pier area. This portion of the image is separated from the rest of the image and further segmented.

The complete segmentation of the pier areas is shown in Figures 62 and 63. The regions in first image are not as clean as the regions in the second since the water in the first image is rougher and sometimes the water blends in with the ship regions. The ships are segmented in both images, but some are only partially segmented and some blend into a part of the piers. The water regions in the second image are clearly segmented and are used for locating the pier area for further segmentation. The shadow regions in the first images are used for the same purpose since the water is not as clearly separable.

4.6.1 Summary of Segmentation

One way to evaluate the advantages of using a "plan" for generation of the segmentation, is a comparison of the time for the segmentation of the complete image with the plan and without the plan. There is no need to segment the full size image to determine the approximate time required for segmentation. We can use the times for the plan generation and multiply them by the size reduction factor (64 for the eight by eight reduction). (Only the times which depend on the image size are multiplied by this factor.) Using the times for the house-2 plan segmentation (Figure 17), we can derive an approximate number of operations for the segmentation of the scene without a plan of 1300.5 million operations, compared with about 226.1 million for the segmentation of the scene with plan generation (see Figure 64). This time difference will more than make up for the extra time required for the generation of the reduced images. With the reduction time and the color transformation times included, the total is about 465.5 million operations. This approximation for the full size segmentation assumes that the times for each operation will increase linearly with the picture size. This is true in the ideal case, but in the current implementation for the very large pictures there is substantial overhead in reading the files from secondary storage.

The segmentation times would not be substantially different if the reduction was by a factor of sixteen rather than a factor of eight, since the expansion of the plan and the reduction of the images accounts for a substantial portion of the total time. Some of the extracted regions are rather "fuzzy" since the plan threshold values are not necessarily the optimal levels for the full size image. The accuracy of the segmentation could be improved with an additional refinement step after the expansion. That is, refinement is performed by the segmentation procedure rather than the plan threshold. In this case the segmentation procedure should concentrate on removing the "tails" from the peak since usually there should be only one peak in each of the histograms.

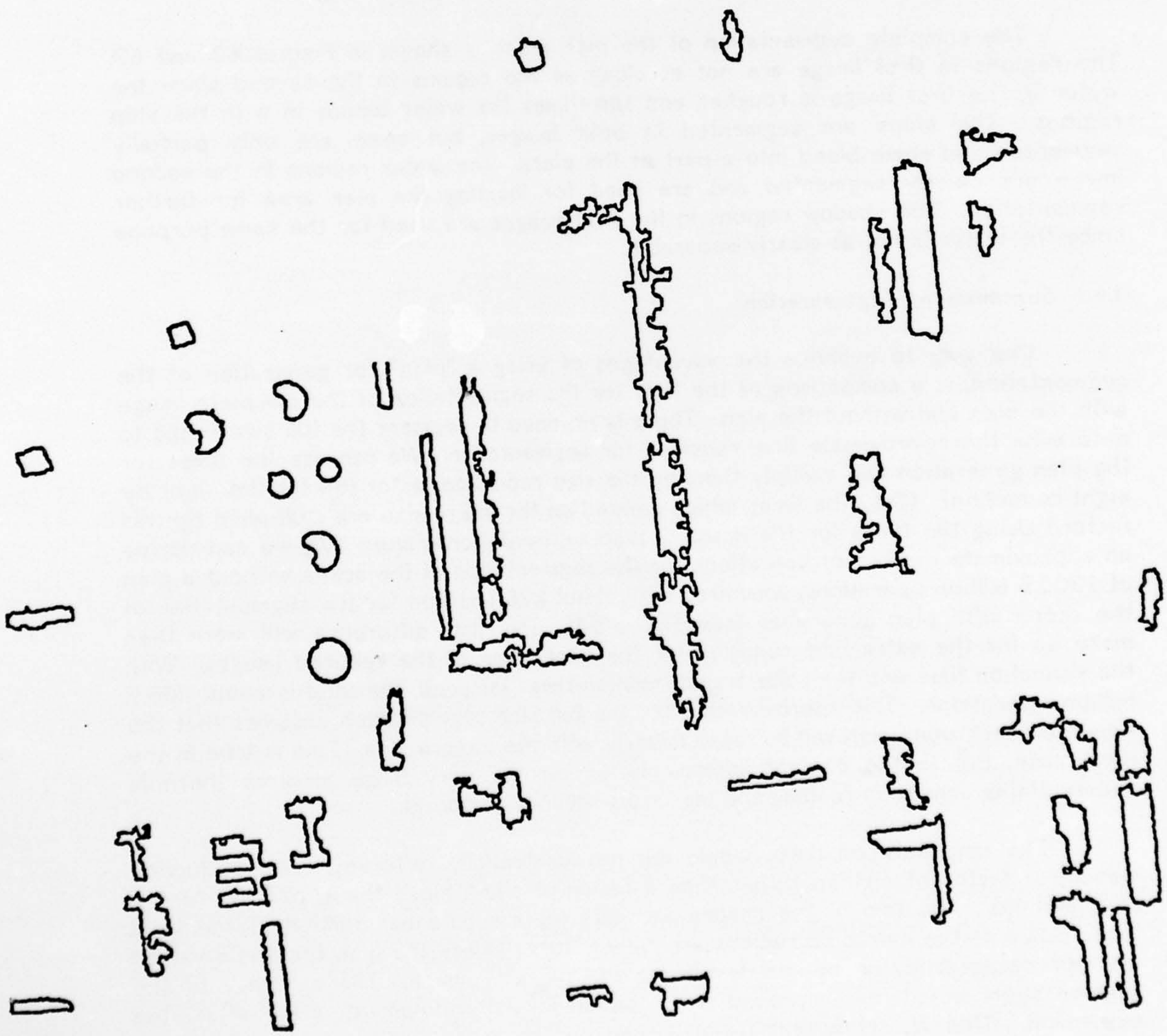


Figure 60 Urban 1 Segmentation

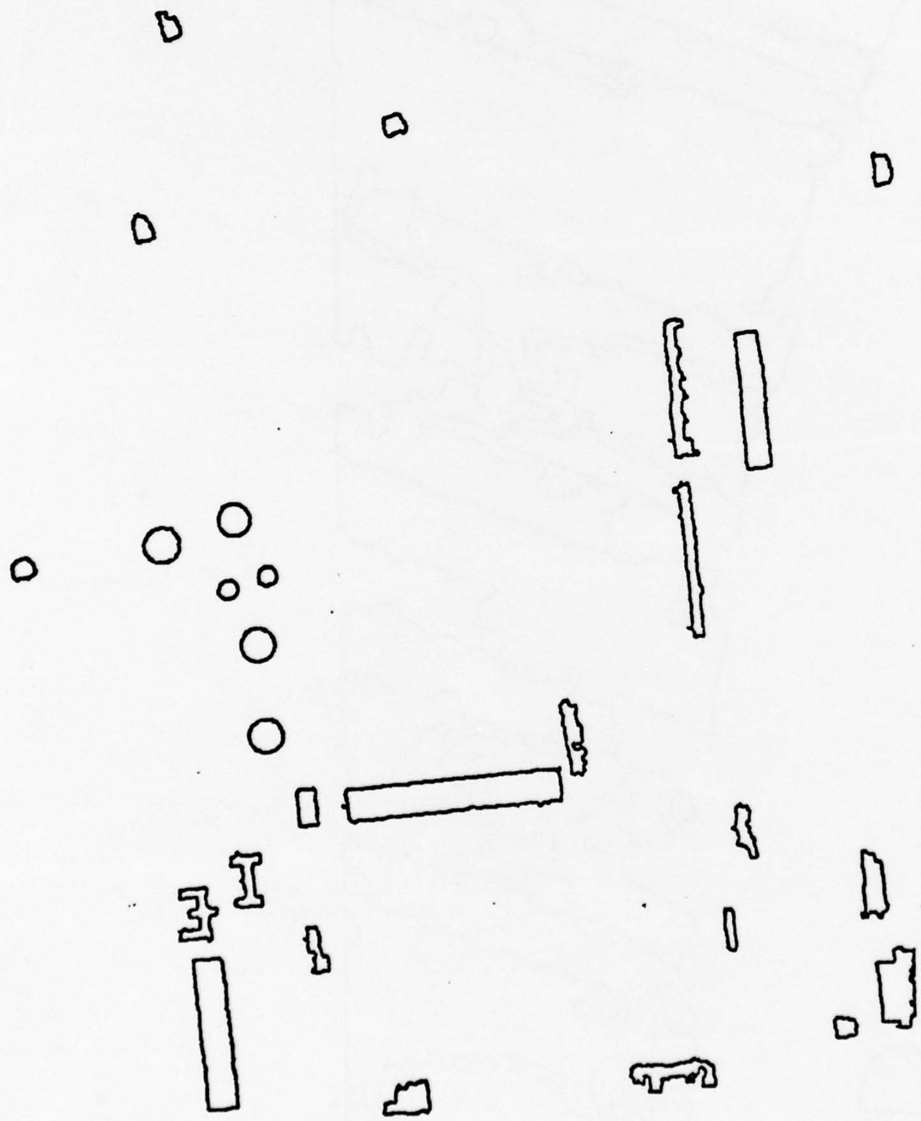
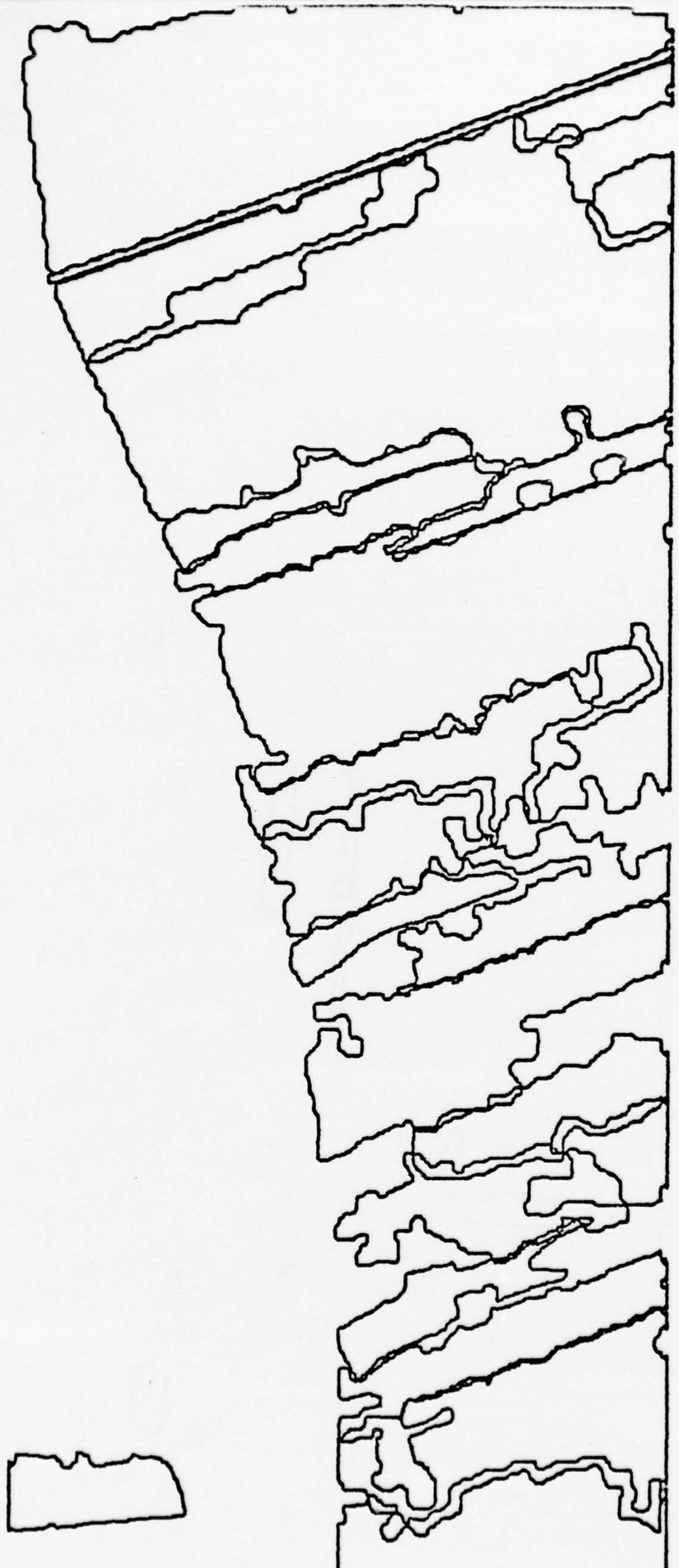


Figure 61 Urban 2 Segmentation



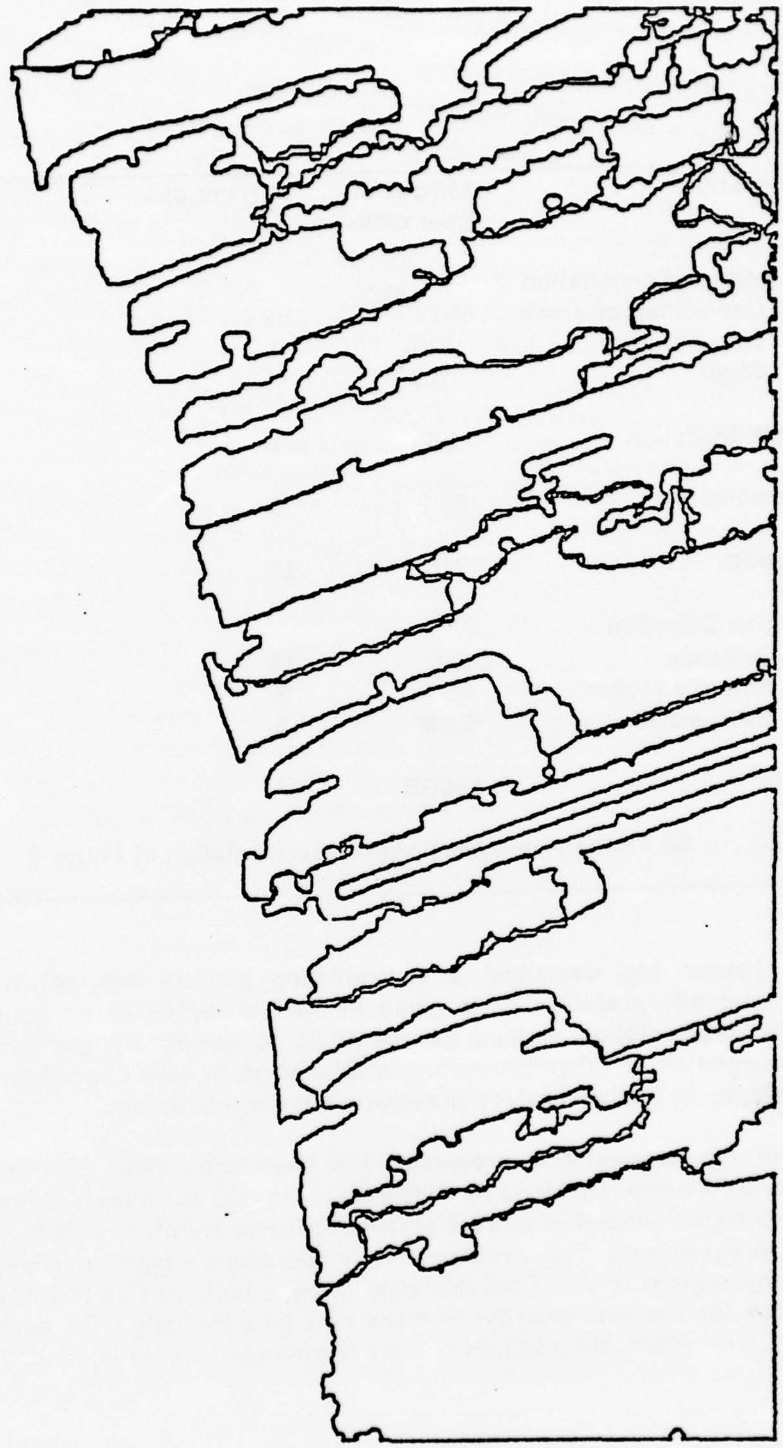


Figure 63 Urban 2 Pier Area Segmentation

Operation	Millions of Operations	Percent of Total
Histogram Computation		
Generation of array	681	52
Smooth array	11.4	1
Other	0.3	--
Peak Selection	8.1	1
Threshold	88.2	6
Smooth	210	16
Region Selection		
Initialize	210	16
Select a region	38.7	3
Save masks	52.8	4
Total	1300.5	

Figure 64 Timing Approximations for Segmentation of House 2

This chapter has described a segmentation scheme and not a combined segmentation and interpretation as is given in Yakimovsky(1973) or Tenenbaum et al.(1976). The segmentation method can be used to completely segment a scene before any recognition or other processing is attempted, to select specific regions for further analysis, or to further analyze previously segmented regions.

Most of the segmentation processing has been automated. The default peak selection criteria (the one described by Ohlander(1975)) has been implemented, but the specific peak selector needed to extract regions indicated by other outside knowledge has not been programmed. This problem also arises when a high priority peak does not segment any regions of an acceptable size: there is no provision to force the peak selection to look for the next priority peak the next time through. This peak selection process is one area where the addition of more knowledge sources is needed.

5 Features

Symbolic analysis of images depends upon the extraction of meaningful features to describe each region. The features can be used for the identification of objects (a task not attempted here), or for the comparison of multiple images to determine changes. Chapter 4 discussed the initial segmentation of an image. These segments (regions) will be used as the basic units for the *extraction of features.* Chapter 6 discusses the use of the features derived here for the matching of regions and images.

This chapter describes the features which were used for segmentation and symbolic analysis. We also discuss the methods of computation of these feature values. Finally we will present some timing information for the computation of many of these features.

5.1 Generation of Features

The features used for the analysis of images will be similiar to those used in understanding images by humans. This will aid in the understanding of extracted features by a human operator and the analysis of the results of this system. These include classes of features such as: size, shape, location, color and texture, and patterns (Akin and Reddy, 1976).

We will discuss the features which we are using under each of these classes (except patterns which would include features such as how many occurrences of an object). We will include some computation times in this section, but the complete timing summary is given in the final section.

5.1.1 Size

The size of a region includes features such as area, length, height, area relative to other regions (largest, smallest), and extent of the region.

The size (area) of the region is just the number of points that the regions covers. This is computed by counting the number of points in the mask which describes the region (either the plan mask, the full size mask, or both). The size of a region is also a by-product of many other feature computations (such as the average intensity) so that the area computation can be considered to take "no" time.

5.1.2 Shape

A human observer describes the shape of a region as irregular or regular (e.g. a rectangle, a circle, a triangle, etc.), elongated, linear, curved, flat, convex, etc.

5.1.2.1 Regular Regions

An <u>irregular region</u> is characterized by having a long perimeter relative to the area of the region, and a small area relative to an enclosing regular object such as a rectangle. The ratios of Perimeter^2/Area and Area/(Area of Minimum Bounding Rectangle (MBR)) (called the "fractional fill") are used for this measure. The perimeter

is computed by the boundary following program given in Appendix 4, and is the number of pixels which are on the outside border of the region. The Perimeter^2/Area measure is chosen rather than Perimeter/Area since it is a dimensionless quantity. (In a continuous world this ratio would be minimal for circles, but this is not necessarily true for the digital world, where there are no true circles. This measure will not distinguish circles and diamonds, but our primary use for the measure is to distinguish compact regions from "loose" regions.) The fractional fill measure is highly orientation dependent: a long, rectangular region has a very small fractional fill ratio when oriented at an angle, but it is near one when the region is horizontal.

5.1.2.2 Elongated Regions

An elongated region is a region with a high length to width ratio; this is also called eccentricity. This length to width ratio can be calculated from the dimensions of the MBR for the region. This method of calculation is simple, but it assumes that the region is oriented in the MBR so that the primary axis of the region is parallel to the longer side of the MBR. Elongated regions will also appear to be irregular regions since Perimeter^2/Area is large for long and thin regions as well.

5.1.2.3 Orientation of Regions

Because of the problems with the simple length to width ratio using the MBR dimensions, it is desirable to obtain an orientation-independent length to width ratio and the orientation as well. In the work by Duda et al.(1972) on the analysis of weather radar images, there was a discussion of the use of Fourier transforms of the boundary for reducing storage requirements of the contour. There was also a mention of some of the properties of the values at various harmonics of the transform. For example, if the contour is reconstructed with only the first harmonic, the new contour appears as an ellipse. The orientation of the major axis of the ellipse, and the ratios of the major and minor axes of the ellipse can be used for the orientation and length to width ratios. We will now present the general techniques used to generate the Fourier coefficients which can then be used to generate these two measures.

5.1.2.3.a Fourier Computations

The contour of the region is represented in terms of two functions I(s) and J(s), which give the I and J coordinates of each element on the contour. These functions are periodic about the contour (i.e. I(s+P)=I(s), where P is the perimeter length) and the reconstruction can be made as accurate as desired by increasing the number of harmonics used.

The formulae to reconstruct the contour from the Fourier coefficients are:

$$I(s)=a_0+\sum_{n=1}^{\infty} a_n \cos(n\,\omega\,s-\theta_n) \qquad (1)$$

and

$$J(s)=b_0+\sum_{n=1}^{\infty} b_n \cos(n\,\omega\,s-\varphi_n) \qquad (2)$$

where a_n and b_n are the amplitudes and θ_n and φ_n and the phase angles of the n^{th} harmonic, and ω is the common fundamental frequency $2\pi/P$ where P is the length of the perimeter.

The Fourier coefficients are given by:

$$a_0 = A_0 / P \tag{3}$$

$$a_n = (2/n\pi) \sin(n\pi/P) \, \mathrm{SQRT}(A_n^2 + B_n^2)$$

$$\theta_n = \tan^{-1}(B_n / A_n) - n\pi/P \tag{4}$$

Where

$$A_n = \sum_{k=1}^{P} I_k \cos(2\pi kn / P) \tag{5}$$

$$B_n = \sum_{k=1}^{P} I_k \sin(2\pi kn / P) \tag{6}$$

Where I_k is the I coordinate of the k^{th} boundary element. With b_n and φ_n defined using equations (3) and (4) by substituting J_k for I_k in equations (5) and (6). The constants a_0 and b_0 are the average of I_k and J_k, the center of mass of the border.

Using polar coordinate we can write the two parametric equations of the ellipse (i.e. the reconstruction with one harmonic) as:

$$I_1(s) = a_0 + a_1 \cos(\omega s - \theta_1) = a_0 + r \cos(\alpha) \tag{7}$$

$$J_1(s) = b_0 + b_1 \cos(\omega s - \varphi_1) = b_0 + r \sin(\alpha) \tag{8}$$

which then gives the equation of the ellipse as:

$$r^2 = \frac{2a_1^2 b_1^2 \sin^2(\theta_1 - \varphi_1)}{a_1^2 + b_1^2 + (b_1^2 - a_1^2)\cos(2\alpha) - 2a_1 b_1 \cos(\theta_1 - \varphi_1)\sin(2\alpha)} \tag{9}$$

The angle of the major axis is given by the relation:

$$\tan 2\beta = \frac{2a_1 b_1 \cos(\theta_1 - \varphi_1)}{a_1^2 - b_1^2} \tag{10}$$

The angle of the minor axis is simply $(\beta + \pi/2)$. These two angles can then be substituted for α in equation (9) to determine the length of the major and minor axes and thus the length-to-width ratio.

Generally the computation of Fourier coefficients is considered to be an expensive operation. But in the application here it is not much more expensive than the computation of the border itself, especially when only the first harmonic is computed. In the house images the mean time for each boundary computation is about 0.3 million operations and the mean for the Fourier coefficient computation (including another boundary computation) is about 2.8 million operations for the first nine harmonics. For fewer than nine harmonics the times are much closer. In another

image where only the first two harmonics were computed the mean for the boundary computation alone is 0.52 million operations and 1.32 million for the Fourier computations (including the 0.52 million for another boundary computation to determine the border coordinates). Since we use only the first harmonic (i.e. two terms: the zeroth and first), the coefficient computation time would be significantly less if the program were designed as a special purpose program to compute these coefficients rather than as a general program to compute any number of coefficients. A more complete timing analysis and discussion is given in the final section of this chapter.

5.1.3 Location

The location of a region includes both the absolute position in the scene, and the position relative to other regions. Position relative to other regions includes features such as above, below, neighboring, to left, to right, etc.

5.1.3.1 Absolute Position

The absolute position features are defined (for our purposes) as the location of the center of mass of the region. The location of the extremes of the mask, or any other consistent location in the mask would also be reasonable. The center of mass for I and J coordinates are used as two separate features. The center of mass is computed as the mean I (and J) coordinate location. The time required for this computation is little more than the time required to compute the size and also gives the size as a necessary by-product, that is about 0.45 million operations for each region in the house-2 image, and size alone would be about 0.19 million operations.

5.1.3.2 Neighbors

Two regions are adjacent if their borders touch (or come close to touching) at some point. The following procedure describes a method to calculate all the neighbor relations for a list of regions:

> For all of the regions in the list do the following:
>
> Follow the boundary of the region: (see Appendix 4 for a description of a boundary following program).
>
> a. Store the outline of the region in a temporary buffer using a unique identifier for the region (e.g. a sequence number).
>
> b. Check the neighborhood of this point (in the temporary buffer) to see if this region is adjacent to any other region which has already been outlined, and, if true, store the adjacency relation.

The neighborhood size is used to determine how close two regions must come to be considered to be adjacent (e.g. a neighborhood of one point on either side of the boundary means that the regions "touch", two points means that there may be at most one pixel between the regions, etc.).

Since we usually calculate the neighbors with the plan results, the times are not excessive: less than 2.3 million operations for each of the house images. A third of that time is for reading the mask buffers from secondary storage. Because of the increased overhead when image buffers will not fit into core memory, this calculation would be very expensive if it was performed on the full size segmentation.

5.1.3.3 Relative Position

Another useful location feature is the position of one region relative to another. Region 1 is <u>above</u> Region 2 if the top of R1 is above the top of R2, the bottom of R1 is above the center of mass of R2, and the regions overlap horizontally. This is expressed as:

$$\begin{aligned}&(\mathrm{Top}(R1) < \mathrm{Top}(R2))\\ \wedge\ &(\mathrm{Bottom}(R1) < \text{Center-of-Mass-I}(R2))\\ \wedge\ &(\mathrm{Left}(R1)\ \mathrm{MAX}\ (\mathrm{Left}(R2)) \le (\mathrm{Right}(R1)\ \mathrm{MIN}\ \mathrm{Right}(R2))).\end{aligned}$$

<u>Below</u>, <u>To-right</u>, and <u>To-left</u> are defined in the equivalent manner. This operation turned out to be one of the more expensive feature computations. Very little of the time doing image calculations; most was in the checking of the four relations between pairs of regions (about 68.53 million operations for the house images, i.e. twice the time for a plan generation).

5.1.4 Color and Texture

The color and texture feature includes all spectral information and transformations of it, such as saturation, intensity, red intensity, which color, what textural pattern. These features are the parameters which are used in the segmentation process.

5.1.4.1 Color

In some of the preceding segmentation examples there were nine spectral features: Red, Blue, Green, Density, Hue, Saturation, Y, I, and Q. The first three are the output of the scanner and are used to generate the last six. Density, Hue, and Saturation are psychologically inspired features and are based on the color triangle (which is a color solid when Density is included) (from Tenenbaum et al., 1974). Y, I, and Q are U. S. color television standards (from Hunt, 1967). The I and Q which we use have been scaled so that the values are positive (Kender, 1976). The formulae for computing these are:

$$\begin{bmatrix}Y\\ I\\ Q\end{bmatrix} = \begin{bmatrix} .299 & .587 & .114\\ 1.0 & -.460 & -.540\\ .403 & -1.0 & .597\end{bmatrix}\begin{bmatrix}\mathrm{Red}\\ \mathrm{Blue}\\ \mathrm{Green}\end{bmatrix} + \begin{bmatrix}0\\ M\\ 0\end{bmatrix}$$

Where M is the maximum value in the image.

Density=(Red + Blue + Green) / 3 + 0.5
(A rounded average is used for the digital representation.)

Hue Computation depends on the sextant of the circle.
Given: Angle=ARCTAN(SQRT(3) * (max - mid)/(max - min + mid - min))
Hue=<for max=r, mid=g, min=b> 60+angle
<max=g, mid=r, min=b> 60+angle
<max=g, mid=b, min=r> 180-angle
<max=b, mid=g, min=r> 180+angle
<max=b, mid=r, min=g> 300-angle
<max=r, mid=b, min=g> 300+angle

$$Saturation=maxsat * (1 - (3 * min) / (Red + Green + Blue)) + 0.5$$

Where maxsat is the maximum saturation value, min is the minimum color value (Red MIN Green MIN Blue), max is the maximum color value (Red MAX Green MAX Blue), and mid is the the middle color value.

These are not the only spectral features that we used. The LANDSAT pictures have four bands ranging from green to infra-red. Other images contained only one band, the density.

The color features for a region are the average (and standard deviation) of each the spectral features over the entire region. For a nonlinear feature such as hue it is possible only to compute the average hue as the hue of the average red, blue, and green.

The figures given in Appendix 3 indicate that these six color transforms should require about 51 operations per pixel. The actual color transform for the reduced images (91 pixels by 93 pixels) required 4.431 million operations, or about 524 operations per pixel. This discrepancy can be partially explained by the fact that each basic operation as counted in Appendix 3 is not necessarily one operation on the current implementation (a high level language on a PDP-10). Also, there is substantial initialization of certain tables to make the transform calculations faster which has a greater impact on the times for these small images than on the times on much larger images.

The extraction of the color feature averages is done on the full size images. The transform colors (density, Y, I, and Q) could be generated from the average red, green, and blue, but are not. Hue and saturation must be computed from the average of the three colors. In the house-2 image, the average number of operations for each color average is 0.489 million (i.e. about the same as the center of mass).

5.1.4.2 Texture

Texture poses a different problem since it is harder to quantify. For computer analysis, texture can be viewed as a statistical or structural property, but for humans textural descriptions are usually structural (e.g. "checker board pattern", "herring bone pattern", "random pattern", "lined", etc.). Such structural descriptions are harder to derive and would primarily contribute to the description of regions. Statistical descriptions offer the best features for incorporation into our general segmentation procedure. There are many different textural descriptions which could be computed, but a few simple measures are sufficient for our purposes, the segmentation applications and some region description applications. These measures are intended to locate regions which are untextured (i.e. homogeneous), or regions of high contrast. When used as symbolic descriptors of a region, these features are used in the same manner as the other "color" features. Rosenfeld(1969) has discussed several texture measures including the use of micro-edges. Haralick et al.(1971) describes several measures which are used to generate a single textural description for a large area of the image.

A common use of the textural measure in segmentation is the location of regions which contain little textural information (i.e. smooth, homogeneous regions). A homogeneous region is one where there are few points which would be selected as an

edge point by some edge operator. Since we are interested in an indication of the possibility of an edge at a point (i.e a micro-edge) rather than of collecting edges into lines and objects, we do not need an accurate edge locator or follower.

A micro-edge should be indicated at a point where the image values are changing (in its neighborhood), but should not be indicated in constant areas or in areas with a constant intensity gradient. The following describes one (of many) possible methods to generate micro-edges. If we look at one row (or column, or diagonal) of the image, we can say that an edge occurs at each point where the derivative of the intensity (with respect to position in the row) changes sign. Since the actual derivative of the image values is not trivial to compute, we can approximate it at each point by the difference between the point value and the one before it.

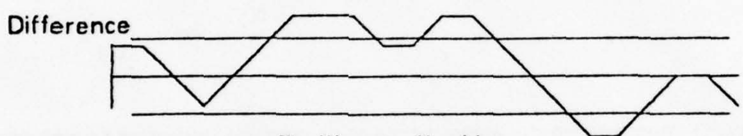


Figure 1 Micro-edge Computation Using Zero Crossings

In Figure 1 edges would be marked at the three points where the difference value crosses the zero line. In general, if all transitions are included there are zero crossings at far too many points: a homogeneous region does not have exactly constant values. Therefore a "noise level" must be used to limit the indicated edges (indicated by the extra horizontal lines in the figure). The "noise level" means that, instead of zero crossings, we are looking for crossings of a band between +noise and -noise. An edge is indicated where the difference goes from above the noise level to below the negative of the noise level (or the reverse). (Thus the initial definition corresponds to a noise level of zero.) With the noise level indicated in the figure there would be only one edge, just to the right of center. The operator is applied in both the horizontal and vertical direction (at the same time) producing a binary image where a point is "1" if a micro-edge is indicated at that point, and "0" if no micro-edge is indicated. Figure 2 shows several edge images of the Urban-1 image with different noise levels. The points where edges are indicated appear black in the figures. This does not necessarily mark all the true edges in the image (no matter the noise level), but the true edges are not the intended result. A less constrained definition of the operator would be to mark a micro-edge when at least one of the extremes is outside the noise range, rather than both. This texture operator takes about 481.6 million operation for the Urban-1 image, or about 118 operations per pixel. It is also possible to generate one complete micro-edge file (for all noise levels) and extract each noise level with a threshold operation, which is how the sequence of noise level figure was generated.

Other textural measures, which also can be used for the generation of the planning image, include the maximum value in the window and the total difference of values in the window (or the excursion of values). The maximum value would indicate areas where bright points occur (possibly a single bright point which would be lost in the mean computation). The minimum value in the window is a similiar operation; it shows up dark points. The excursion image shows the areas of the image where there

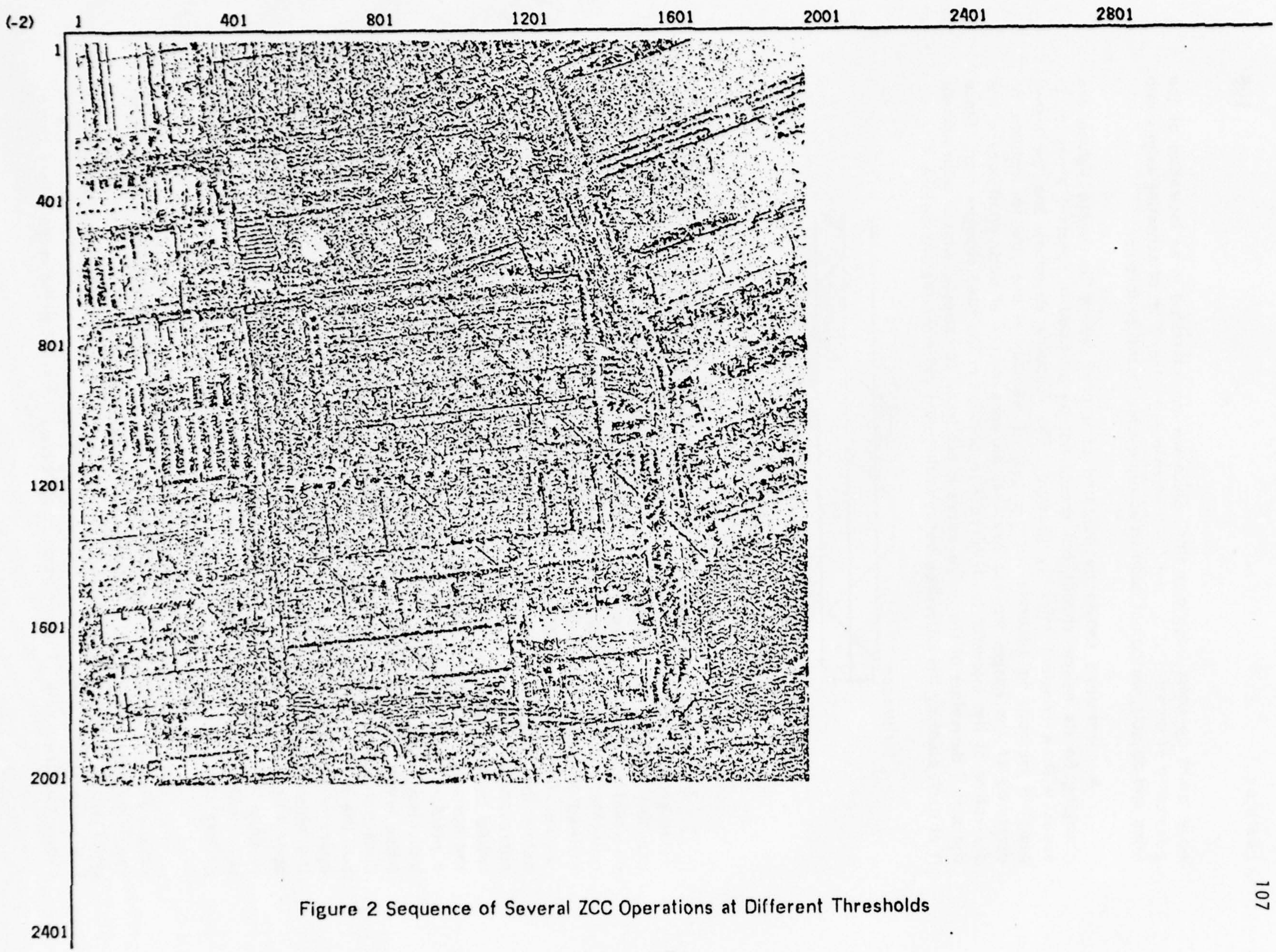
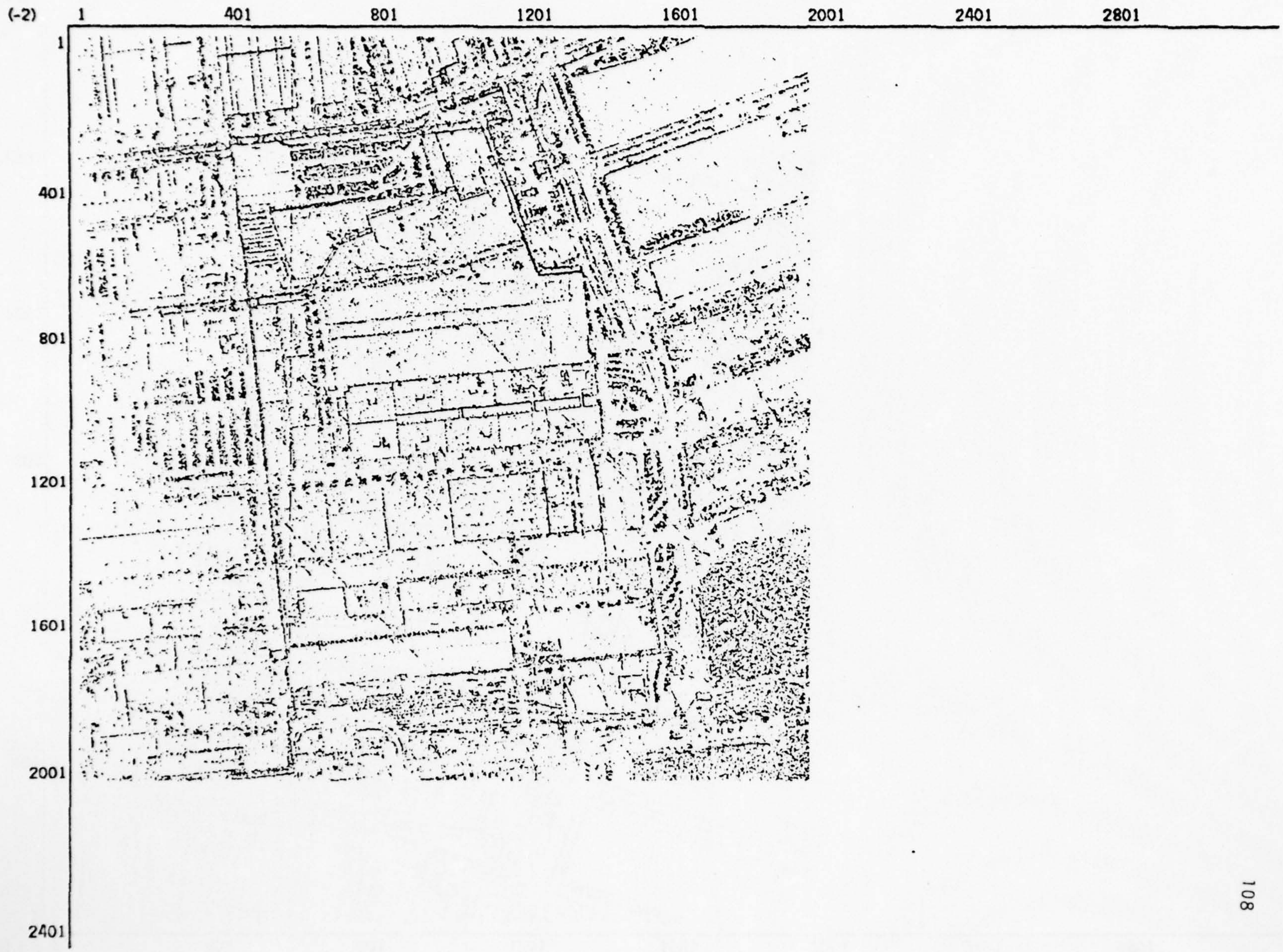


Figure 2 Sequence of Several ZCC Operations at Different Thresholds

N1Z 5-∞


(-2)
1
401
801
1201
1601
2001
2401
2801
1
401
801
1201
1601
2001
2401

NIZ 9-∞


(-2)
1
401
801
1201
1601
2001
2401
2801
1
401
801
1201
1601
2001
2401

NIZ 12-∞


(-2)
1
401
801
1201
1601
2001
2401
2801
1
401
801
1201
1601
2001
2401

N1Z 15-∞

NIZ 19 TO INF

are large (or small) changes in the reduction window: high contrast areas or low contrast areas. The maximum and minimum values are a necessary by-product of the computation of the excursion value. These two were used with limited success for the SLR images where textural separations were desired. These operations take about the same number of operations as the reduction procedures (78.3 million for Images the size of the house and cityscape scenes).

Another textural measure is the variance in the reduction window. This value is generated along with the mean by the reduction program. We used this feature in the matching in the color images, but not for segmentation.

5.2 The use of Features

We selected a large group of features to describe an image so that its description could be compared with other images of the same scene. This is covered in detail in the next chapter (Chapter 6). In addition, these are the same classes of features that would be needed in a system designed to analyze and recognize features in a single image.

5.3 Results

Feature Computation	Millions of Operations	Number of Times Used	Mean Number of Operations
Neighbors	2.37		
Read files	0.74	22	0.034
Follow outline	1.62	22	0.074
Relative Positions	68.53	496	0.138
Compute Center Mass	11.37	25	0.455
Extract Center Mass	6.02	674	0.009
Color (9) Average	140.98	288	0.489
Count (size)	4.52	24	0.188
Border length	7.23	24	0.301
Shape Transforms (9)	67.73	24	2.822
Orientation	1.39	50	0.028
Length to Width	0.08	50	0.002
Variance	0.97	25	0.039
Save Data	5.47		
Read Data	8.19		

Figure 3 Feature Extraction Times House 2

Figures 3, 4, and 5 show the feature extraction times for the second house image, the first cityscape image, and the second Urban pier regions. None of the individual operations is expensive when taken alone, but when the feature computation is applied to many regions, some can be expensive. The relative position

Feature Computation	Millions of Operations	Number of Times Used	Mean Number of Operations
Neighbors	3.44		
Read files	1.34	31	0.044
Follow outline	2.06	31	0.067
Relative Positions	119.85	946	0.127
Compute Center Mass	11.95	30	0.398
Extract Center Mass	8.74	1087	0.008
Color (9) Average	215.53	348	0.619
Count (size)	7.29	29	0.251
Border length	11.28	29	0.389
Shape Transforms (9)	93.92	29	3.239
Orientation	1.78	60	0.030
Length to Width	0.09	60	0.001
Variance	4.33	30	0.144
Save Data	6.71		
Read Data	21.89		

Figure 4 Feature Extraction Times Cityscape 1

Feature Computation	Millions of Operations	Number of Times Used	Mean Number of Operations
Neighbors	6.98		
Read files	1.51	28	0.054
Follow outline	5.47	28	0.195
Relative Positions	77.63	496	0.157
Extract Center Mass	0.16	756	----
Color (2) Average	53.28	52	1.025
Count (size)	9.67	26	0.372
Border length	11.01	26	0.423
Shape Transforms (2)	34.57	26	1.330
Orientation	0.05	54	0.001
Length to Width	0.08	54	0.001
Save Data	3.65		
Read Data	5.15		

Figure 5 Feature Extraction Times Urban Pier 2 Subsection

calculation (above, below, etc.) compares each region in the image with all other regions (except the regions already compared since the relation is reflexive) so that a relatively cheap computation, the checking for relative position relations between two regions, becomes expensive because of the many calls. Each individual relative

position operation takes a lot of time to retrieve much of the information from the LEAP data base every time, and LEAP is not the most efficient storage mechanism.

The operations such as size, center of mass, and color averages all require about the same number of operations. For these features, the expense is in looking at the picture points (or mask points) rather than the feature computations. Some of the features are generated from the results of other operators (such as P^2/Area, fractional fill, and orientation), and, therefore, are very cheap (the time is in the procedure overhead and several LEAP operations to extract the feature values).

In terms of the total time required, the expensive operations are the reduction, color, and texture computations. Most of the initial operations (color etc.) and the feature operations could be performed on much simpler (i.e. cheaper) special purpose (or even general purpose) processors since the limiting factor on the feature computation speed is the time required to read through the the image rather than the computation at each point. An exception to this is the relative position computation, which could be improved by storing the position information more efficiently for this program, instead of using the general LEAP storage facility. These feature extraction times represent very unoptimized implementations, and do not reflect the best attainable times. Since each single application of the feature extraction operators took so little time (as shown in the column giving the mean number of operations per application), little effort was applied to making these operations more efficient.

6 Matching and Change Detection

Change detection has many uses. Among them are the following: Analysis of changes in objects in a scene or in the scene itself. Analysis of stereo images for precise location and altitudes. Precise registration of images taken at different times or from different sensors. Analysis of medical data to detect health changes. But before change detection and analysis is possible, it is necessary to match images or parts of images. This chapter explores the use of features (see Chapter 5) in matching regions of images, and the analysis of changes in the images. We explore some of the problems with current systems for change detection, and propose the use of symbolic analysis to avoid these problems.

In this chapter we will present the symbolic registration and change analysis methods. We will begin with a simple example of symbolic matching. This example will be used to illustrate the basic technique, and to point out some trouble spots for correlation-base registration systems. We will then discuss some of the aspects of changes in certain features, and illustrate the extraction and use of these features in our work. The last section will present the results of the symbolic registration processing for the six scenes and a discussion of the time required for this final matching operation.

6.1 Matching of Regions

The above change detection problems all require a preliminary step of locating correspondences between the parts of the image. Earlier systems used correlation measures (or similar match measures) to find matches for many pairs of points, and then warped the entire picture to minimize the differences in these matching pairs. This matching process will fail when the area being matched is obscured in one image, or when the selected point is in the middle of a homogeneous region, where it will match almost any point in the corresponding region as strongly as any other. Also, warping the image will not work when objects are in different relative positions in the two images.

Our method is to find corresponding pairs of regions in the two images (called *symbolic registration*) using the features discussed in Chapter 5. The selection of which features should be used in the matching process, and the determination of which of these features are most important for the task being considered is controlled by the semantic knowledge. The guiding knowledge includes what the task is and which features may or may not change.

6.1.1 Matching Example

We will first present two simple examples of the operation of symbolic matching. These examples will be used to show some of the problems encountered by correlation-type registration systems. The examples will also illustrate the basic techniques used in symbolic registration to avoid these same problems. Consider the two simple "images" shown in Figure 1, which have the (nonobvious) features given in Figure 2. If we assume that the task is to find the region which best matches region1 in image1, it is clear that region6 in image2 matches region1 using every available feature (location, size, length to width ratio, neighbors, color, etc.). No

other region matches with all these features (region4 differs in color, and region5 differs in length to width ratio and size). Therefore, any method of combining feature matches to generate a region-to-region comparison should indicate that region1 and region6 are corresponding regions. The same holds true for matching region2: the best match is region4, but for region3 the best match of region5 differs in the absolute position feature. This difference is less than the differences between region3 and region4, or regino3 and region6, so the best match for region3 should be region5.

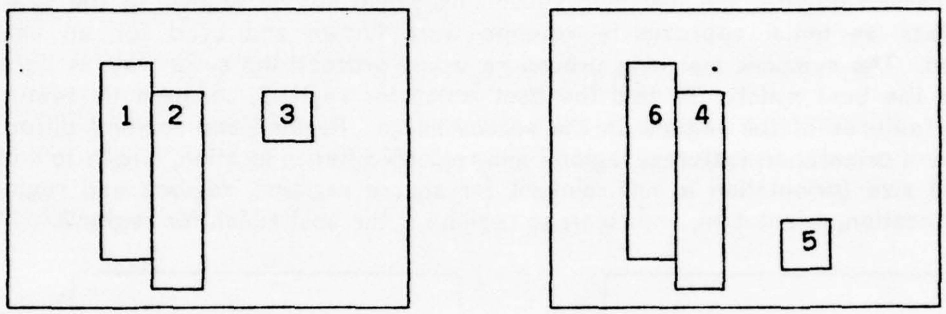


Image 1 Image 2

Figure 1 Simple Match Example Regions

Image 1				Image 2			
color of	1 is	red		color of	4 is	blue	
color of	2 is	blue		color of	5 is	blue	
color of	3 is	blue		color of	6 is	red	
L/W of	1 is	4		L/W of	4 is	4	
L/W of	2 is	4		L/W of	5 is	1	
L/W of	3 is	1		L/W of	6 is	4	

Figure 2 Simple Match Example Properties

In this first simple example a correlation-type matching program should perform well on the left side of the image where region1 and region2 match region6 and region4 with no relative position changes, but the position difference between region3 and region5 could cause problems in determining any global warping transformations. Correlation-based registration schemes should generate an area where an object is missing at the location of region3 and an area where a new object appeared at the location of region5 rather than just indicating that region3 moved (which should be the result for symbolic registration). This means that correlation registration works well under some conditions (few changes in the relative positions of objects) and less well under others (changes in the position of objects when compared to other objects). But, symbolic registration programs should work under both of these conditions.

Consider the two images in Figure 3 (these two images are rotated ninety

degrees with respect to each other). These two images will present more difficult problems for correlation matching and searching procedures. Unless an initial approximation of the relative rotations is known, the search for matching points will be complicated. For example, the system described by Lillestrand(1972) and Allen et al.(1973) assumes that the orientation of the two images to be compared is approximately the same. Their procedures scan across the two images computing the warping functions for each subsection of the image as the matching points are found. With extreme rotations, the matching subsection would not be appear in the search area unless an initial approximate rotation were known and used for an image correction. The symbolic matching procedure would proceed the same way as before to locate the best match. To find the best match for region2, compare its features with the features of the regions in the second image. Region2 and region4 differ in location and orientation features; region2 and region5 differ in location, length to width ratio, and size (orientation is not relevant for square regions); region2 and region6 differ in location, orientation, and color; so region4 is the best match for region2.

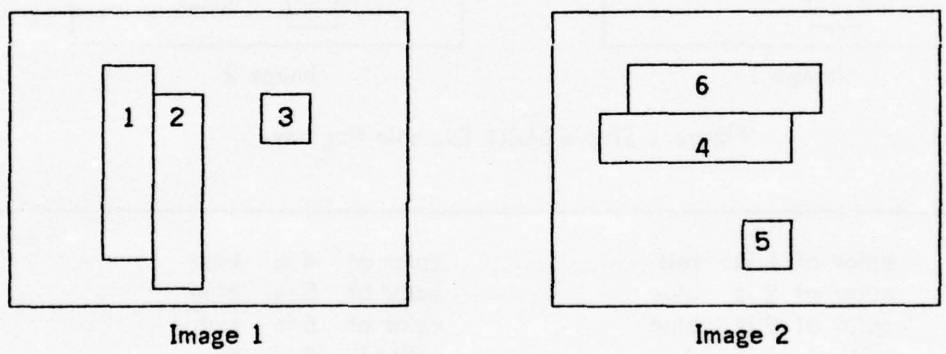


Figure 3 Second Match Example Regions

This last example shows that some features can be more useful than other features, if the knowledge given for the task specifies which features will probably change and which are more reliable. If, in these examples, it is given that there may be an orientation change or a position change, then these two features (and any related features) would not be used in the matching process (or they would receive much less weight in combining their results with other features). In this case, region2 and region4 would match using all the important features; likewise region1 with region6, and region3 with region5.

6.1.2 Region Matching

In a set of real images, the question is usually not whether the regions match exactly, using a given feature, but how close the regions match. When applied to the initial matching problem (finding the corresponding regions), the question is: how well do the two regions match compared to how well the first region matches other regions in the second image using all features - both those that should not change and, to a lesser extent, those that can. Once two regions are known to be corresponding regions, they can be compared again, with the same procedure, to determine which features have changed between the two images, and how much the features have changed. We would like a general purpose program which could be used for both of

these operations: finding the best match, and performing a region-to-region comparison to find changes. This procedure would produce both an indication of how well the regions matched with all the features, and an indication of how well each feature matched. The procedure also must be able to treat features in several different ways: that is, some as very important and constant features, some are probably constant, and some are probably changing. Also, since the important question is how <u>well</u> one region matches another region, we do not want a procedure which generates a complete image-to-image match for all available regions.

When looking for the quality of match between two regions (as when searching for the best match), the rating for the region to region match should be a function of all the feature to feature matches for the two regions. A first order feature to feature match rating is simply the difference in the feature values. But when these ratings are combined, it is necessary to weight the differences due to different features so that each feature has approximately the same contribution to the matching procedure. The feature weights are selected to minimize the effect of near misses since few feature values can be expected to be exactly the same in different images. Some of the weighting values depend on the feature value of the first region, such as the size and average color values. Figure 4 gives the current feature weights. Generally, the feature weight is the inverse of the acceptable difference between the feature values in the two images for the two regions to be considered to match reasonably well. These were arrived at through some experimentation. First a weight was chosen, then several matching operations were performed using this weight. If the matches were good, then the weight was not changed. But if this feature caused many incorrect matches (such as mismatches caused by minor changes in the feature value downgrading the match rating), then the feature weight was reduced so that it would have less effect. Rarely did we make the feature weights more strict.

Feature Type	Inverse of Weighting Value	Comments
Size	Size_of_first*0.2	minimum 100 for plan, 10000 for full size
Colors	$2*\sigma_{color}$	from the color in the first image
Location	12 for plan, 100 for full size image	
Neighbors	1	
Relative Positions	1	
P^2/Area	Value of the first*0.5	
Variance	$\sigma_{feature}$	feature value from the first image
Orientation	0.5	value in radians from $-\pi/2$ to $+\pi/2$
Length to Width	0.5	value from 0 to 1
Fractional Fill	Value*0.3	feature value from first image, 0 to 1

Figure 4 Feature Weighting Values

This gives a feature to feature match rating of:

$$-|V1_i - V2_i| * W_i$$

where $V1_i$ is the value of the i^{th} feature for the region in the first image, $V2_i$ is the feature value in the second image, and W_i is the feature weight. This rating function means that an exact match has a rating of zero, and that the rating decreases as the difference between the values of the features increases. As has been mentioned before, depending on the scene, some features should be weighted more strongly than others when being used for finding corresponding regions. This can be incorporated in the above rating function by adding another term - the strength term:

$$-|V1_i - V2_i| * W_i * S_i$$

Where S_i is the strength of the i^{th} feature. Then the overall rating for the region-to-region match is the sum of all the feature to feature matches. Currently we have three different strength factors for the features, but usually use only two. The different strength functions were chosen so that a poor match using an important feature would out weigh several poor matches on the other, less important, features. Values of 200, 100, and 10 were chosen, but only the lower two are generally used. These matching methods can be used for features with numeric values (such as size, absolute location, orientation, etc.).

But other properties have nonnumeric values. For example, the neighbor_of feature is a relation between regions in the same image. The use of this feature in matching must be somewhat different than the use of the numeric features. It is defined as follows: If Region_1 in the first image has a neighbor Region_X, and Region_X is known to be the corresponding region for Region_Y in the second image, and Region_Y is a neighbor of Region_2, then Region_1 and Region_2 match with the neighbor_of feature. An alternative way to express this is in SAIL:

```
foreach Y such that region_in_next ⊗ (neighbor ⊗ region_1) ≡ bind Y do
        if neighbor ⊗ Y ≡ region_2 then it_is_a_match
                                    else no_match_yet;
```

In this program segment, the regions match if the procedure it_is_a_match is executed (at least once) and fail to match if only no_match_yet is executed. But if neither routine is called, then no judgment can be made, since none of the neighbors of region_1 have yet been matched. The other relations between regions such as above, below, to_left, and to_right are treated in the same way. If the two regions match with these features, then the rating will be zero. If they fail to match, then the rating will be minus the strength value. And, if no judgment is possible it will be minus half the weighting value. (In this last case, it does not matter since no judgment will be made for any pairs checked in the search for a corresponding region.)

6.1.3 Symbolic Registration

The above region matching procedure can be used to determine the quality of a match between region, or the quality of a match between each feature. This procedure is used in the symbolic registration procedure to find the best matching region from a set of potential matching regions.

The symbolic registration procedure is given the following: a region to find the corresponding region, a list of regions (i.e. the second image), several (three) lists of features with each list indicating which features are to be used with different weights,

and the current "best" rating. The program matches the given region with each region in the list using all of the features in the several feature lists. The program stores the best matching region that is encountered, and this region is considered the corresponding region. We also keep track of the second best match which is found, so that we can compare it with the best match to see what features were used to distinguish the best match from the second best. The current best match rating parameter is used to terminate the feature comparison in a region to region match if the match rating falls below this value, since this particular region to region match will never be the best. Since we are locating the second best match, this value should be the current second best match value.

This registration process is mostly automatic. The selection of which features will be given which strengths is based on the expected changes in the images. The user may either select a region and ask the system to produce the best match, or may allow the system to find the best match for each of the regions either in order of segmentation or in order of size. The normal use of the registration procedure is to use it to lacate the corresponding region for a specific user selected region.

The results of using this procedure are given in the last section of this chapter. Appendix 6 gives detailed results for applying this matching procedure on many of the matching regions. Some contain errors. Most are correct matches. These listings give the contributions of each feature to each match, and the mean and standard deviation of the contributions of each feature for all the best matches in that pair of images. This same summary for the second best matches is also included.

6.2 Change Detection

The uses of change detection mentioned in the introduction all require the analysis of changes in some feature value. For example, the detection of changes in medical data usually requires locating objects in one image which were not in the other image. Registration of different images requires the computation of accurate differences in the location of objects in the two images. The analysis of stereo images is similar: finding the location difference for corresponding regions. The analysis of two images in order to find changes in objects in the scene is best done by matching the regions and finding which features changed between the two images.

6.2.1 Kinds of Changes

Since we are concerned with changes in features, we will next study what kinds of changes are possible in the several features.

6.2.1.1 Size

Changes in size occur because of distortions introduced by changes in the camera positions, or by changes in the relative positions of two regions causing one to obscure the other (caused by object or camera movement), or by actual growth (or shrinkage) in the object, or, possibly, by differences in the segmentation of the two images.

The size is greatly altered by changes in the distance of the observer from the

scene (or object), but if the camera positions are not extremely different, most of the larger (smaller) regions in one image will match the larger (smaller) regions in the other image. The effects of a different observer distance can be minimized by adjusting the computed values of the sizes of regions in one image to account for this distance change. This adjustment process is valid only for sets of images where it is given that there is no perspective difference between the two images. When the size adjustment factor is not known, then the size changes from a computed match can be used as an approximation of this factor.

In the urban scene the regions in the first image are larger than the corresponding regions in the second image. When two regions are matched without using size as an important feature, then the difference in size of these two regions are used to adjust the values of the size feature of the regions in the second image for the future matches. Size can then be used as one of the important features. In the urban scene we matched the regions marked "M" (Figure 4.60 and 4.61) without using size as an important feature. The size of region "M" in the first image is 1.484 times the size of region "M" in the second image. This size adjustment factor is used for all future matches between the urban-1 and urban-2 images.

Another size change example is the pair of LANDSAT images, one task is to determine the change in the size of the snow areas. The change due to satellite positions (altitudes) is minimal and the major change is due to the melting. The large middle snow region in the LANDSAT-1 image ("G" in Figure 4.51) is matched with the corresponding snow region in the LANDSAT-2 image ("G" in Figure 4.52) even though the sizes differ greatly (both are the largest region in their respective images). The region in the first image (taken in late May) has 627045 points, and in the second image (taken in early June) it has 354184 points.

The cityscape images are affected by size changes due to different amounts of occlusion of objects, with few size changes due to changes in the objects (the pictures were taken within minutes of each other).

6.2.1.2 Shape

Shape changes are caused by the same factors that cause changes in size, e.g. camera and object movement, growth, and segmentation differences. In some sets of images the camera positions are known to be approximately the same, and therefore changes in shape will be due to changes in the objects. Shape can then be used as a feature in matching. In the two LANDSAT images (the segmentation given in Figure 4.51 and 4.52), the shape (as given by P^2/Area, fractional fill, and length to width ratio) of the snow regions changes due to melting, but the shape of other regions (such as the lakes) remains about the same. For the largest snow region another shape feature, the orientation, does not change significantly.

The rural scene (Figure 4.55,4.56, and 4.57) has orientation changes. The orientation change also means that the regions which are on the edge of the image will be altered in size and shape. It would be possible to use the computed orientation changes to adjust orientation, location, etc in the future matches, but this was not used beyond the adjustment of the orientation.

6.2.1.3 Location

Location changes are caused by the same factors as size changes. But there are additional factors involved in location changes. In oblique views (such as the house and cityscape scene), objects in the scene have different relative positions as the observer position changes. These relative positions changes are due to the different distance from the object to the observer (i.e. a parallax shift). These changes are used to calculate depths from stereo views. Location changes are also caused by actual movement of objects in the scene.

If the location differences are uniform throughout the image (e.g. in the SLR scene; the urban after scale differences are removed; the LANDSAT scene approximately) then the location difference computed for one pair of matching regions (or several pairs) can be used in future region matches in the same way as size differences are used. (Note however that these differences are additive, and size differences are multiplicative.)

In the SLR scene (Figure 4.53 and 4.54) the location difference from good matches of homogeneous regions (those labeled "C") can be used to adjust the locations of future matches. This allows absolute location to be used in the later matches which means that the regions labeled "A" and "B" can be indicated as matching regions.

6.2.1.4 Color and Texture

Color and texture changes can be caused by actual changes in the scene (such as changes in crops), or by lighting differences (a different time of day means that the sun angle will be different and thus shadows will be different), or by sensor or film effects (quality control). Also, no matter how much control is exercised the two views of the same scene will always have minor differences in spectral values. Correlation based change detection systems produce an indication of changes in the spectral values (i.e. color), but these systems require further analysis to deduce that these changes are changes in a region of the image rather than many "random" points. The house and cityscape images had some small differences in the color properties of the various regions. But these differences were not significant enough to affect the matching procedure.

6.2.1.5 Quantity

Changes in quantity are a slightly different problem, since the number of occurrences of an object (or type of object) must be determined before changes in the number are computed. This is the basic task required for the urban scene: find the changes in the number of "ships" between the two images.

As an approximation, we selected a few sample regions in the two images to serve as "ships", "water", and "piers" in an ad hoc image to use for matching. The two "real" images were the matched with this ad hoc image to locate regions which would match to "ships". The regions labeled "S" in the two segmented images (Figures 5 and 6) were matched to the "ship" regions ("S2" means matched to the "two adjacent ship" region). By counting the number of regions matched to "ships" it is possible to determine the change in the number of "ships" in the scene (9 in the first and 21 in

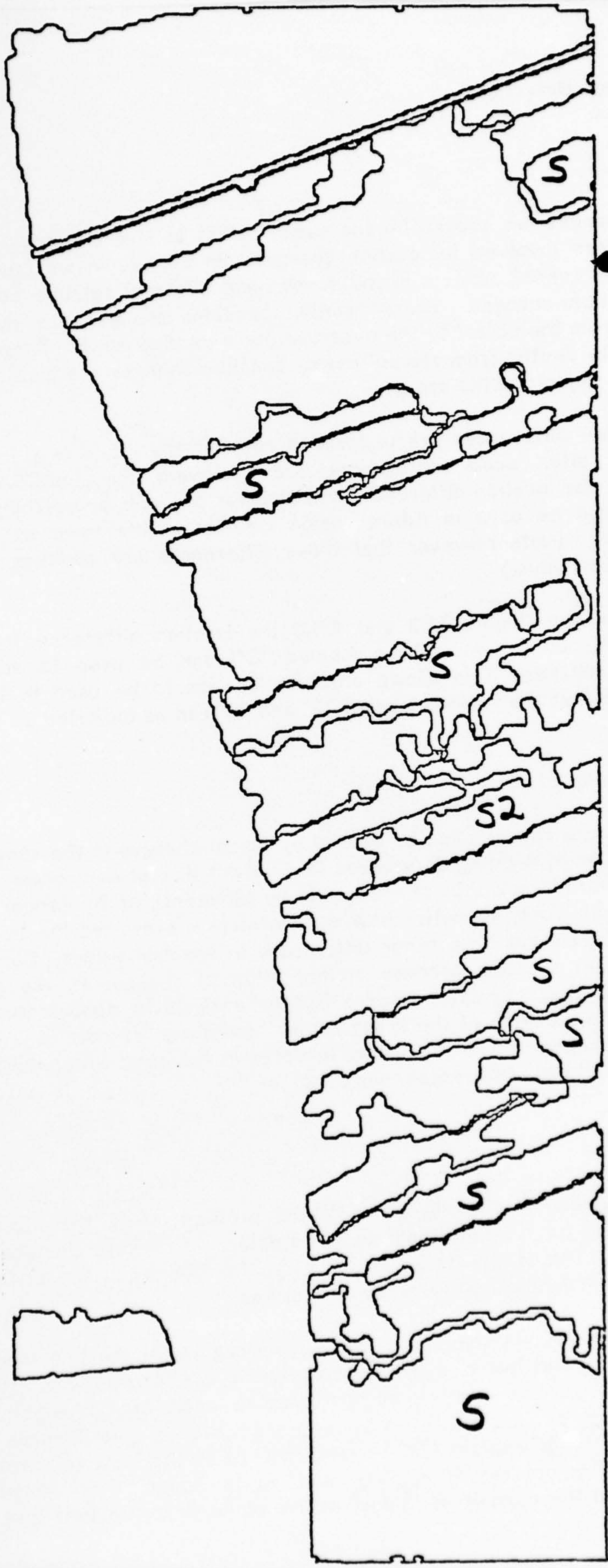


Figure 5 Segmentation of First Pier Area

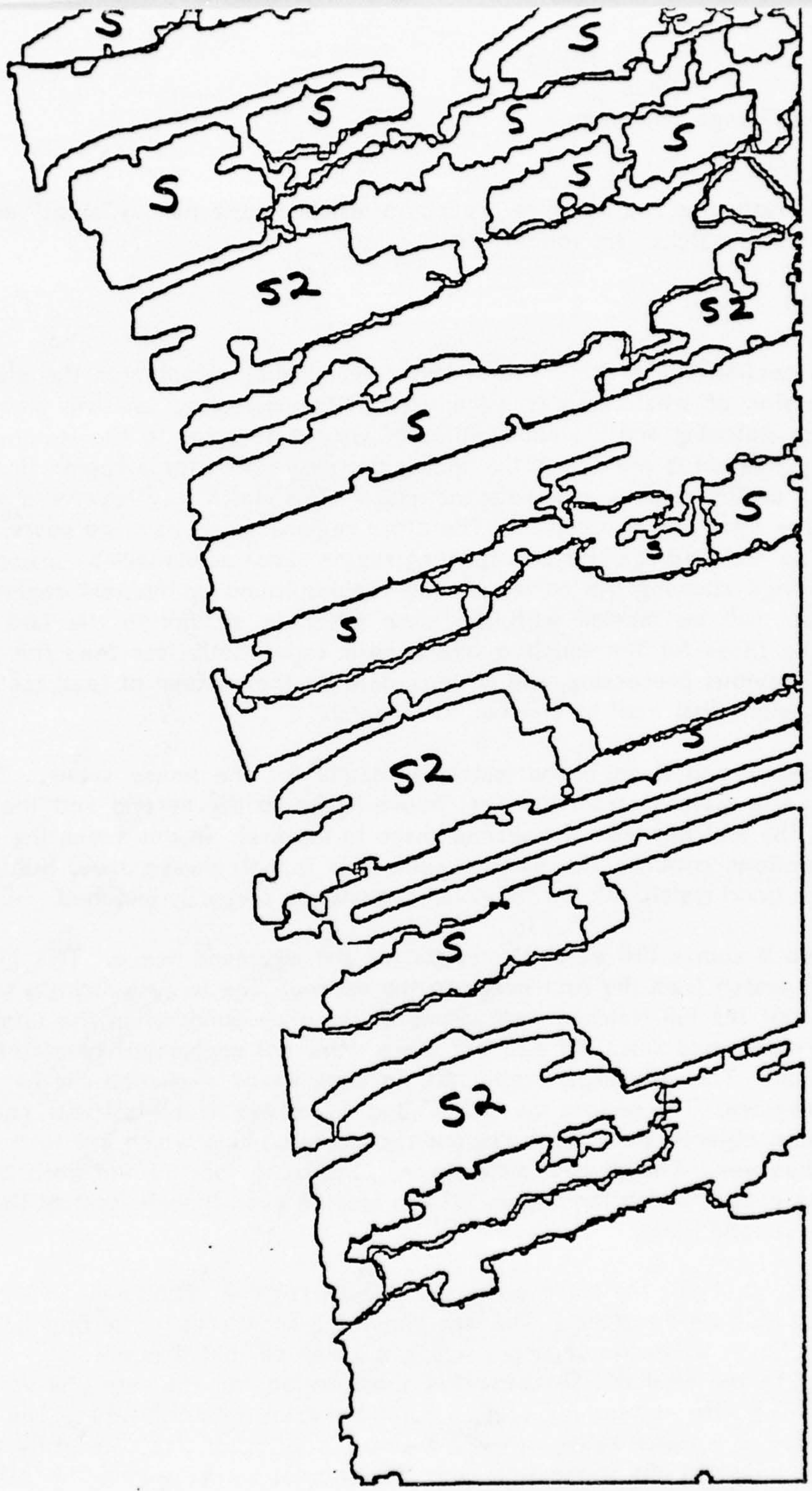


Figure 6 Segmentation of Second Pier Area

the second). Note that the matching procedure matches some partial "ships" as "ships" since other likely matches are too different.

6.3 Results

This section will present the symbolic registration results for the six scenes and a discussion of what features were used in the matching. We will also discuss errors in the matching and the contribution of various features to the matching. The symbolic registration is performed by finding the best match for a region in the first image among all the regions in the second image. This match may receive a very low rating, but the best one is accepted. Therefore regions which have no corresponding region may be matched to some unsuspecting region. Each scene will be presented by outline drawings showing the corresponding regions found in the two regions. The matching pair will be labeled with the same letter or symbol in the two images. Generally, the times for the matching operation is significantly less than the time for any of the previous processing, and is dependent on the number of features and the number of regions that must be checked for a match.

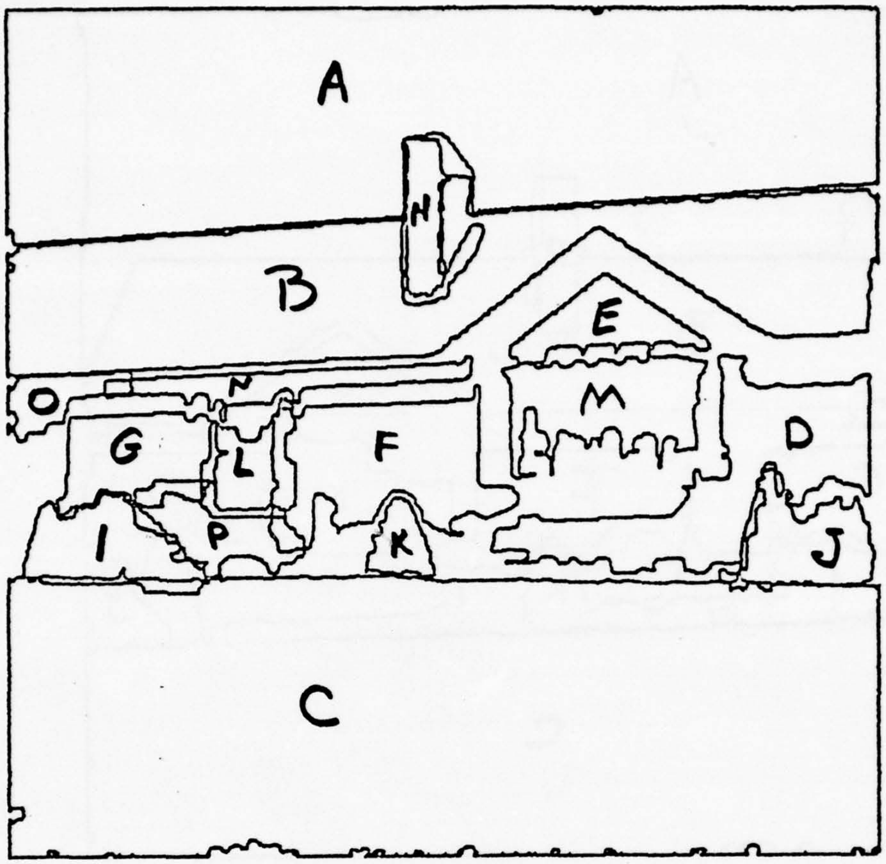
Figures 7 and 8 show the matching results for the house scene. The first figure gives the matches from the first house image to the second and the second figure gives the matches from the second image to the first. In this scene the matches in both directions produce the same results; this is not always true, but it is an indicator of a good match. All the "obvious" regions are correctly matched.

Figure 9 shows the matching results for the cityscape scene. This gives the results for a match from the first image to the second. The large buildings, the park, and portions of the hill matched well. Some of the gray buildings in the upper right matched correctly and some did not, but these were not segmented consistently and are all similiar. This cityscape scene has an example of a change in the relative position of regions. The regions labeled "F" and "G" moved to the left with respect to the foreground objects, since these regions represent objects which are farther away from the observer. The proper matching regions were located for both of these regions. The proper match for region "G" was located even though most of the object is out of the second image.

Figure 10 shows the corresponding LANDSAT regions. The lakes and one snow region are all that were matched. The lake above the snow area in the first image was covered by clouds when the second image was taken so that it is missing. This lake was matched to the smaller lake below the snow region, but the rating is very poor: -1493 compared with -12 for the proper match between the small lakes. The lake on the lower left had a match rating of -177 due to the different size caused by the lake being on the edge of the image (this was the poorest rating among the other lake matches).

Figure 11 gives the matches for the nontextured regions in the SLR scene. The other regions are very vague, and it is unclear what any other matching regions in these images would mean.

For the rural scene, we have six sets of matching results, image one with image three, three with one, two with three, etc. (Figures 12 through 17). The larger



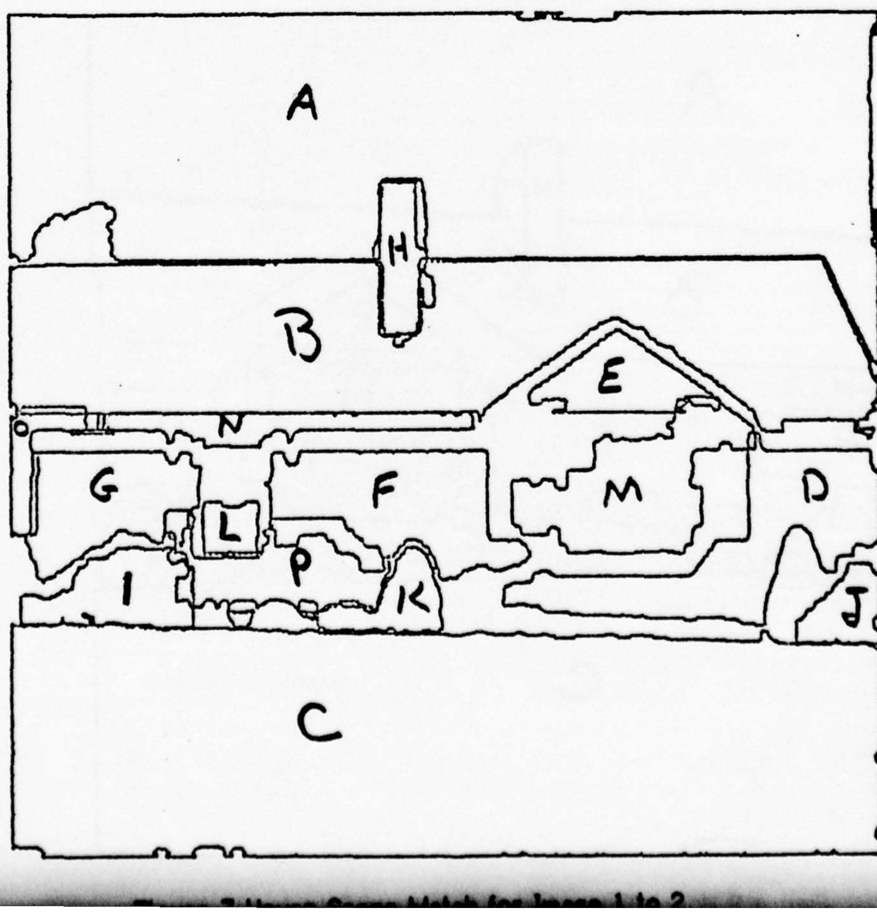
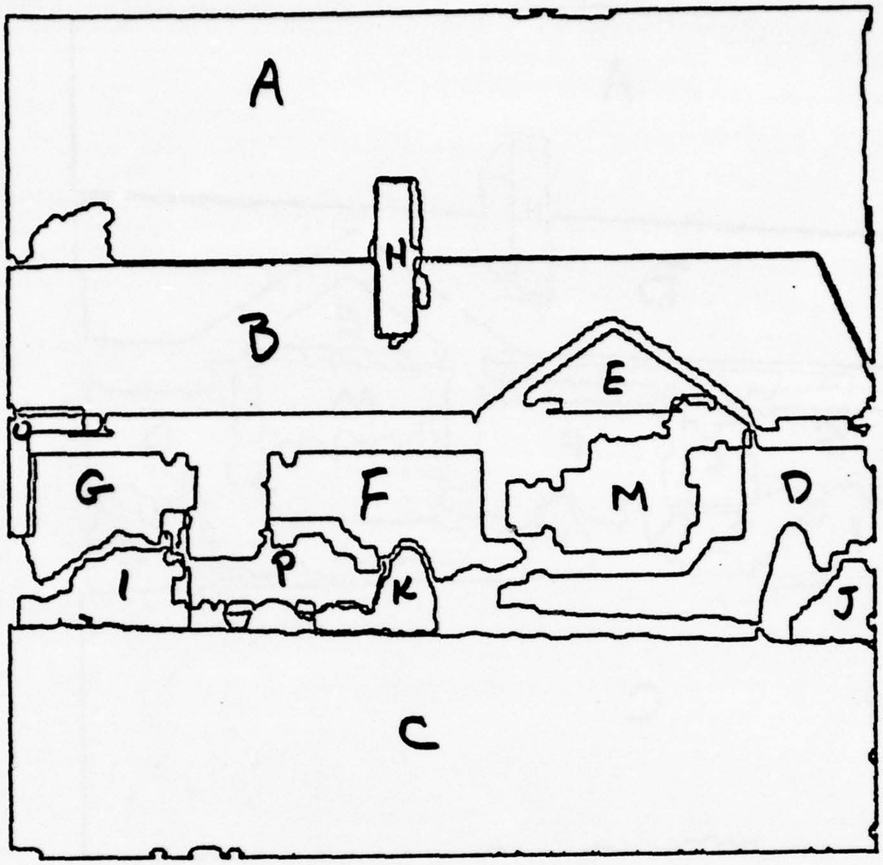
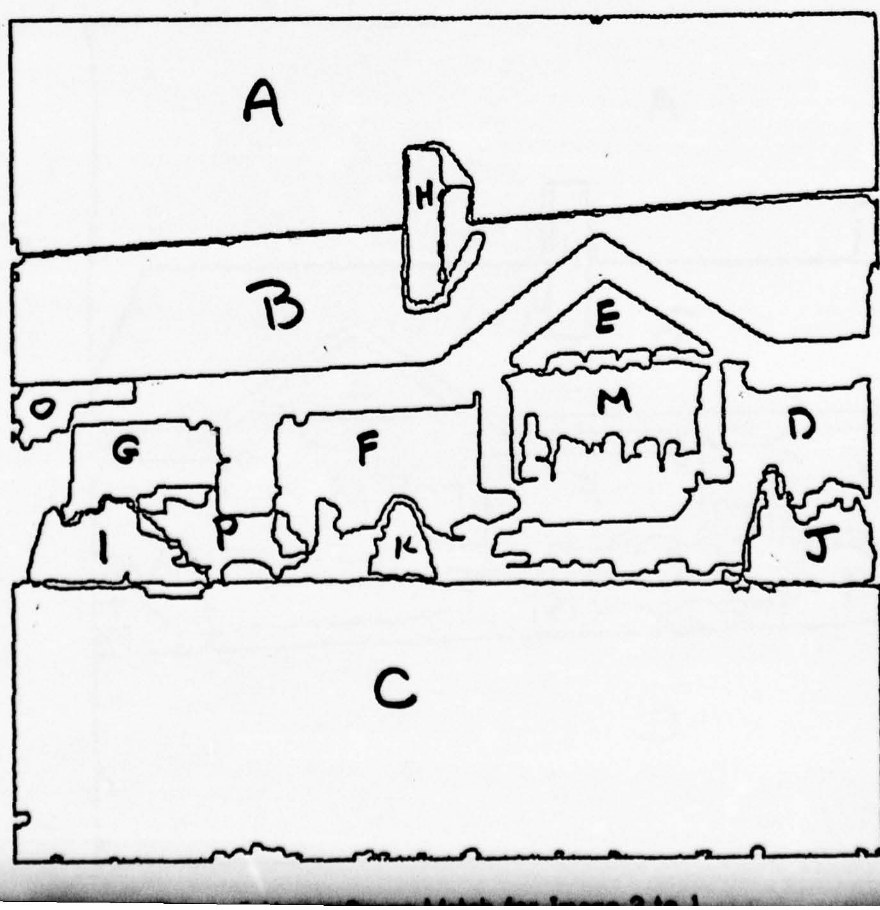


Figure 7 House Scene Match for Image 1 to 2


A
H
B
E
G
F
M
D
P
I
K
J
C


A
H
B
E
O
M
G
F
D
I
P
K
J
C

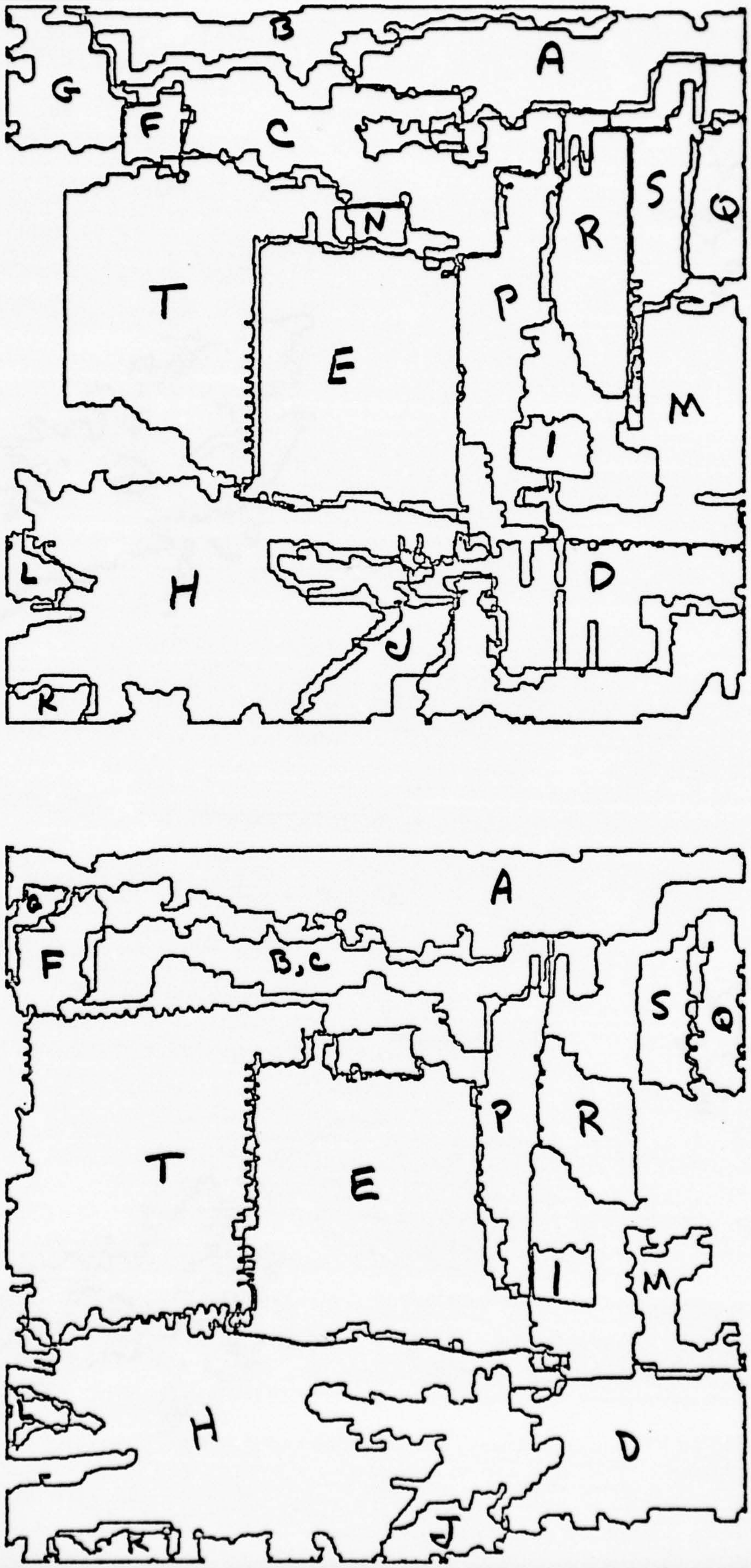


Figure 9 Cityscape Scene Match for Image 1 to 2


A
E
F
G
B
D
C


A
E
G
B
D, F
C

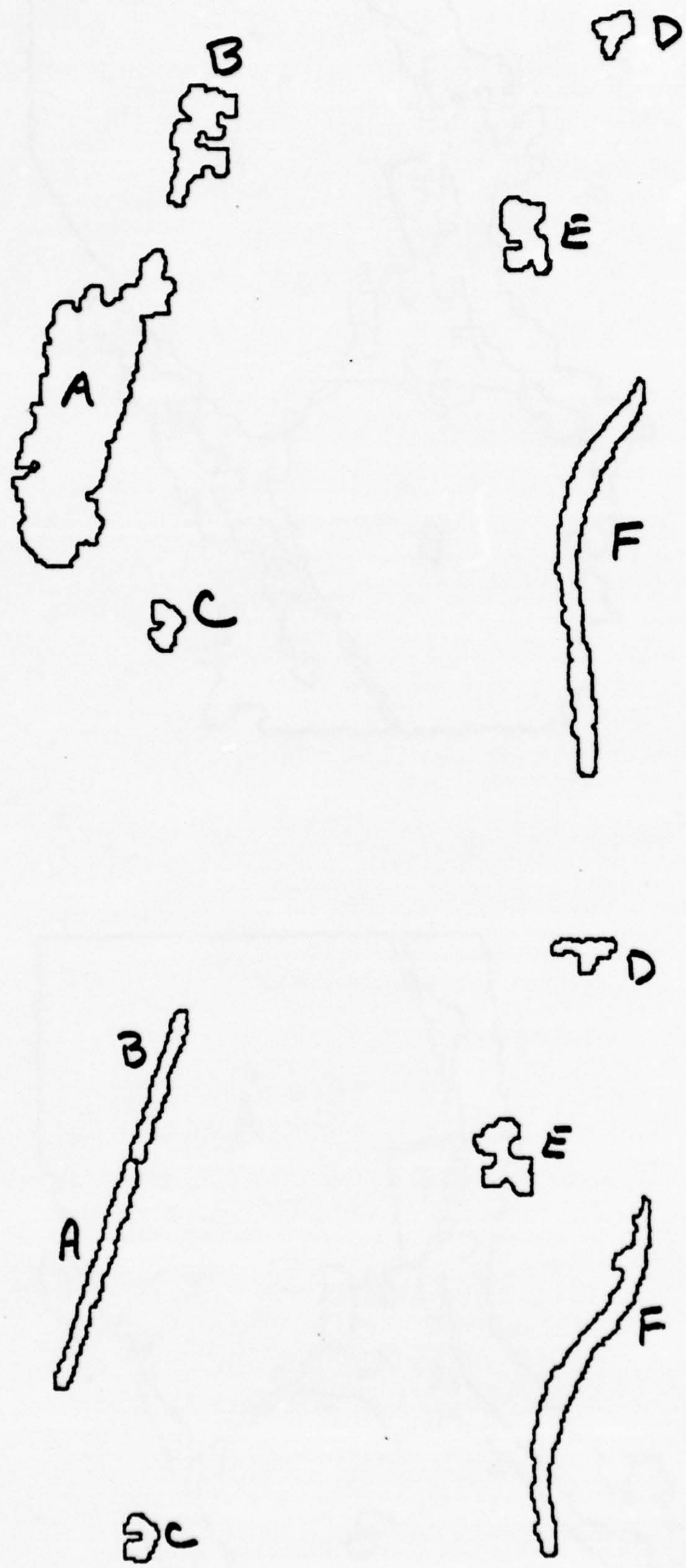
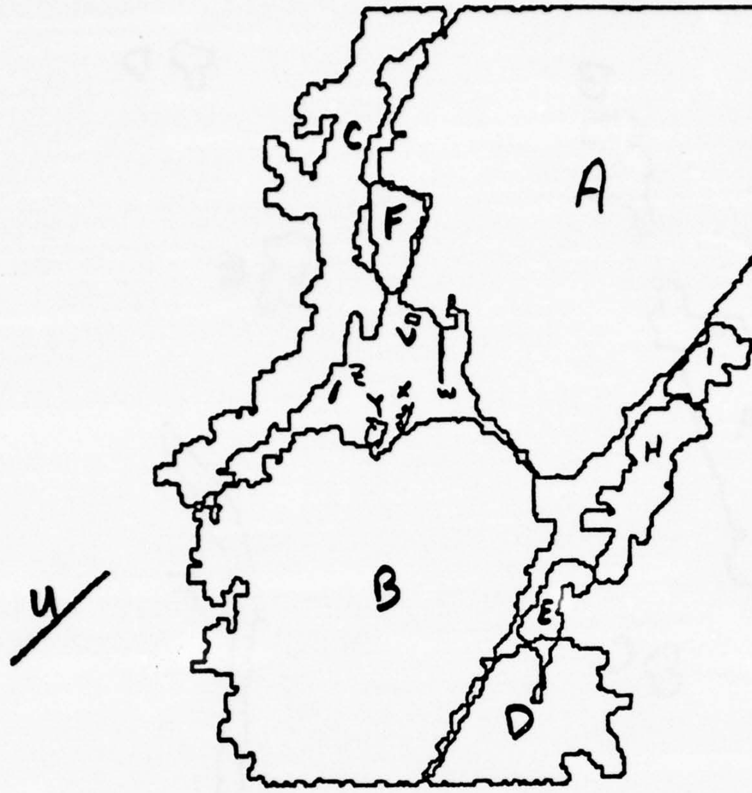


Figure 11 SLR Scene Match for Image 1 to 2



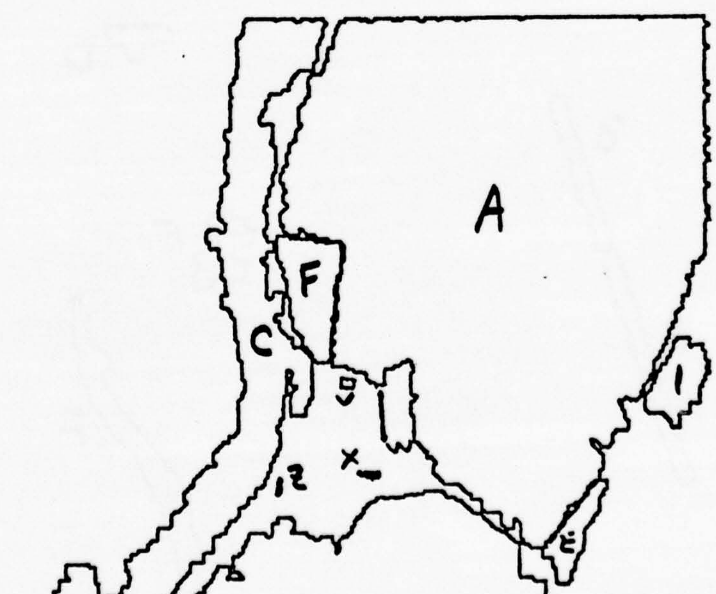
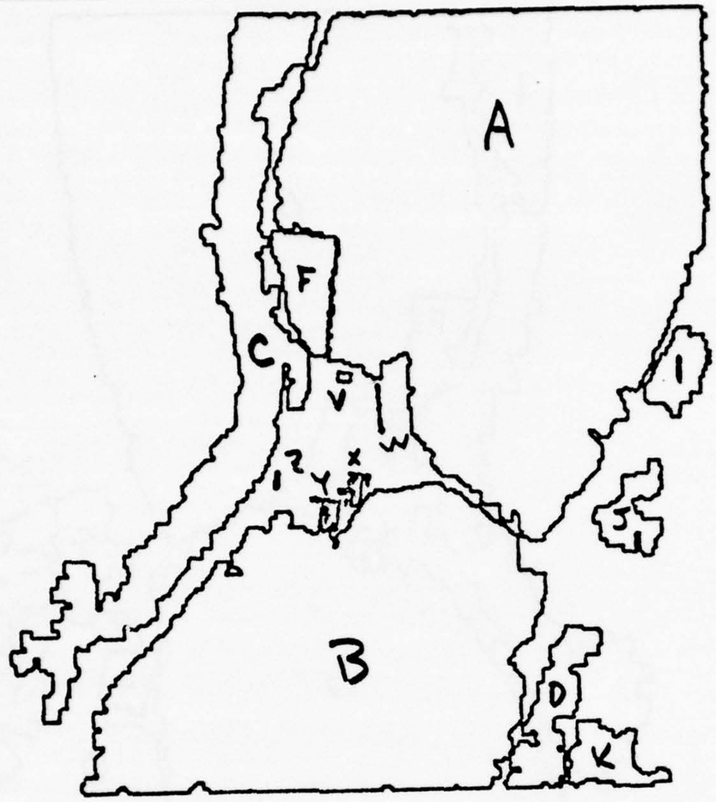


Figure 12 Rural Scene Match for 1 to 3



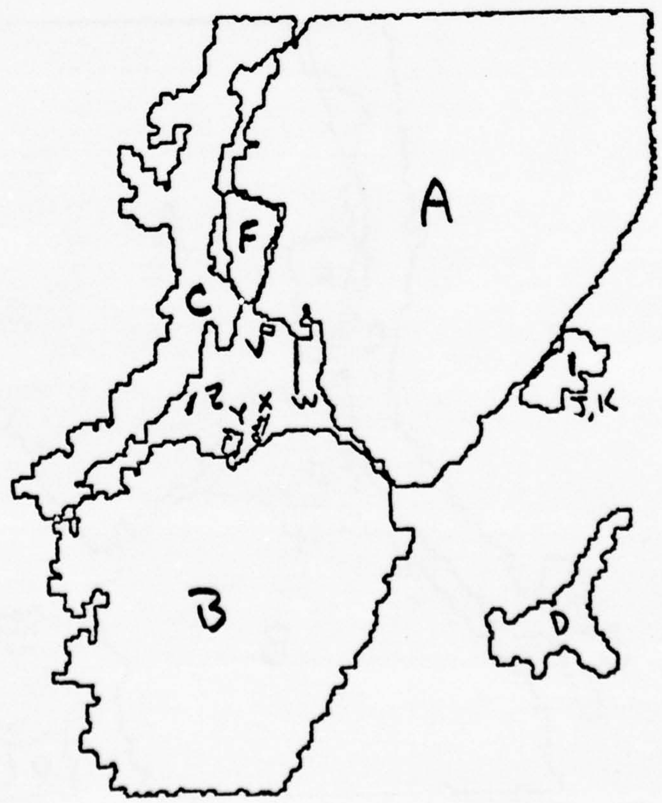


Figure 13 Rural Scene Match for 3 to 1

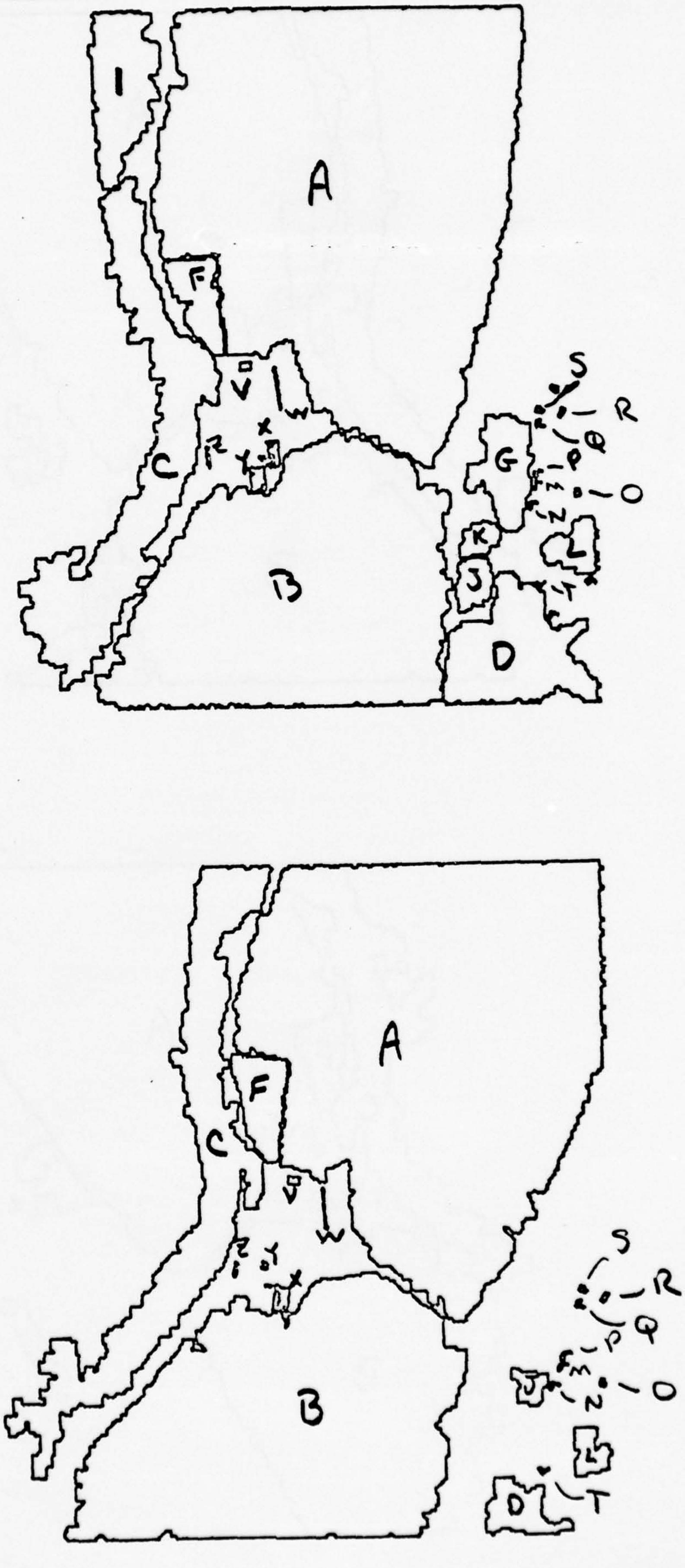
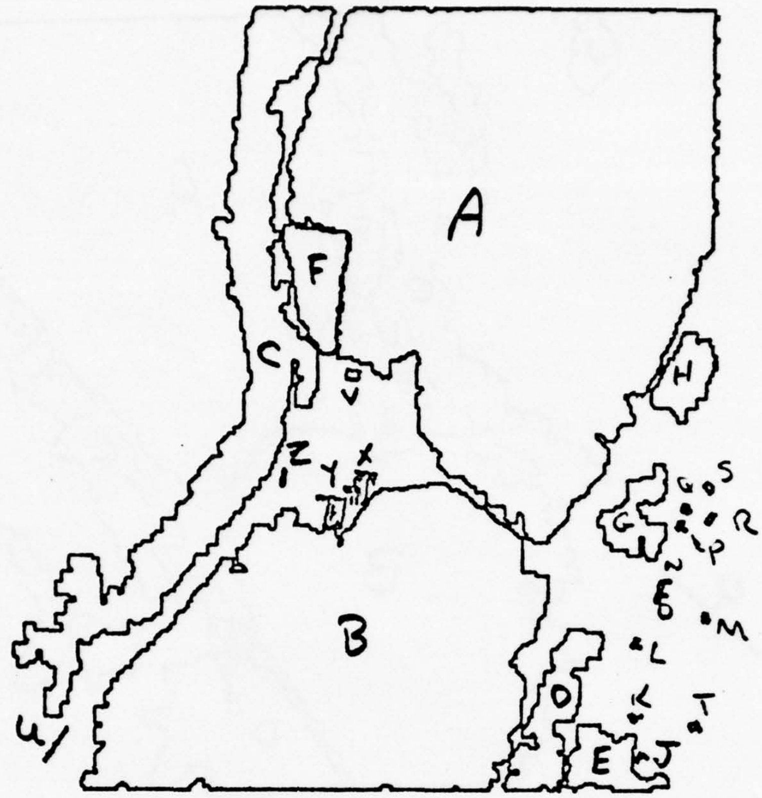


Figure 14 Rural Scene Match for 2 to 3



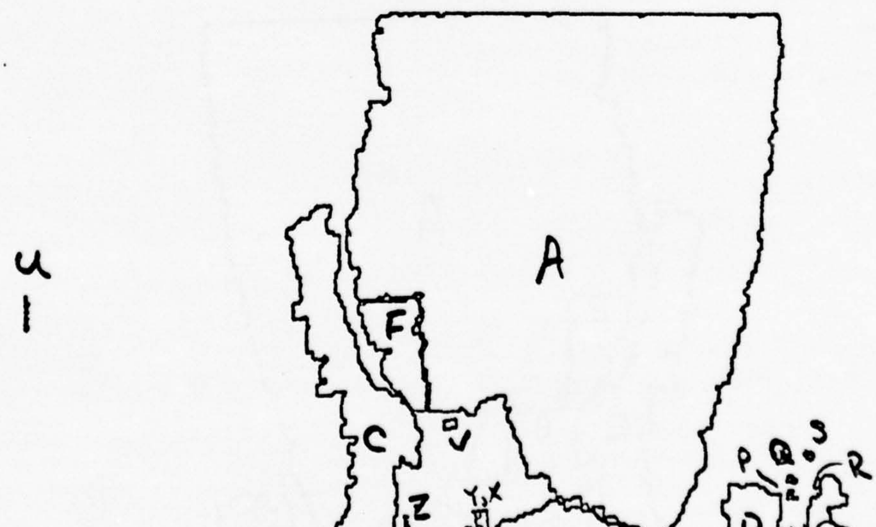
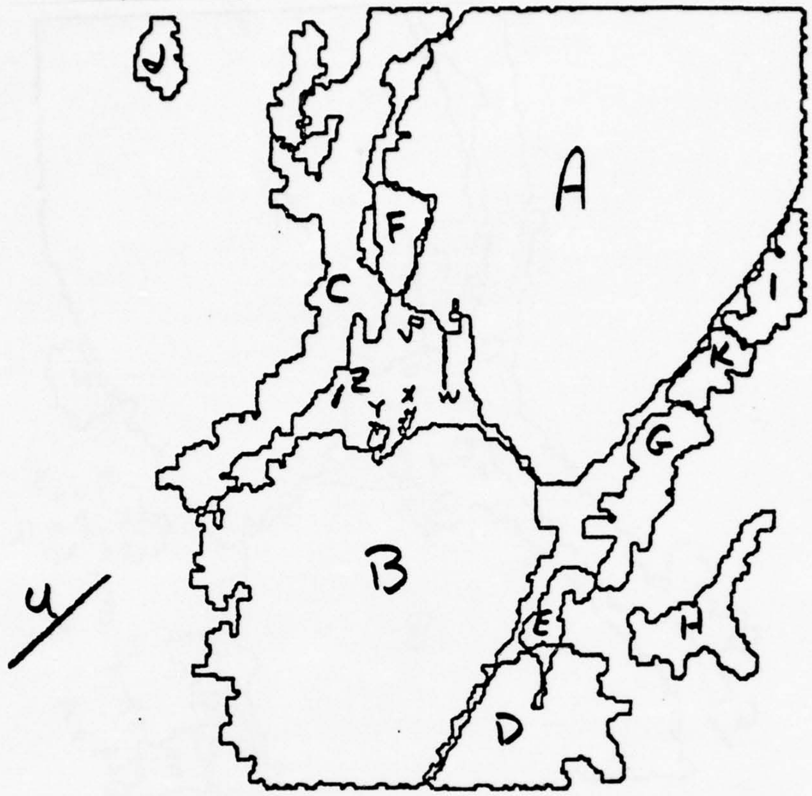


Figure 15 Rural Scene Match for 3 to 2



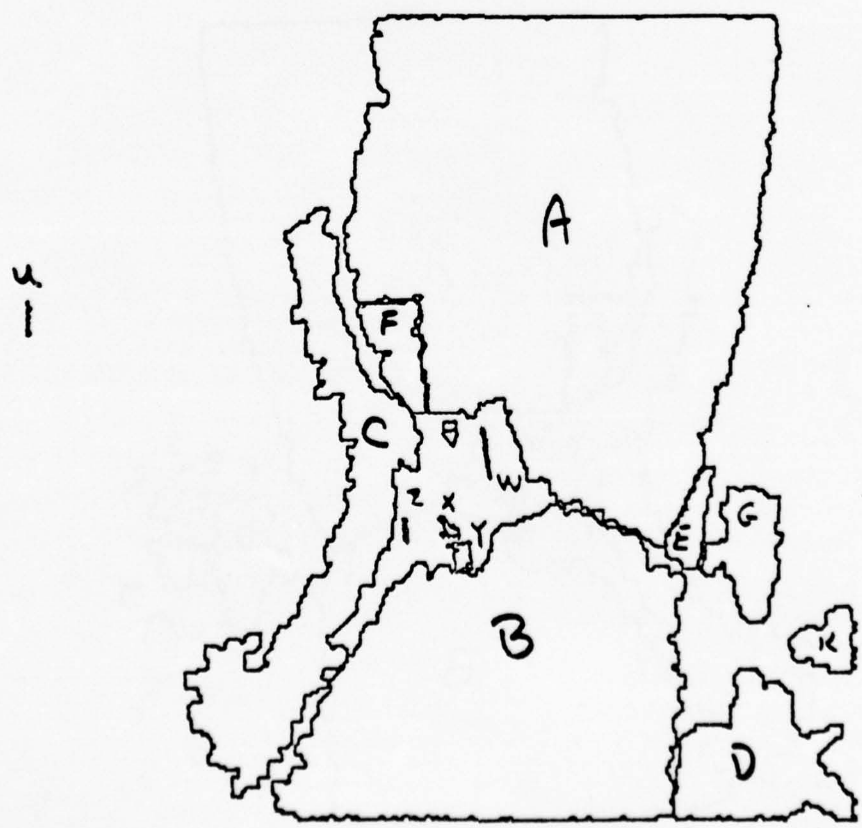


Figure 16 Rural Scene Match for 1 to 2

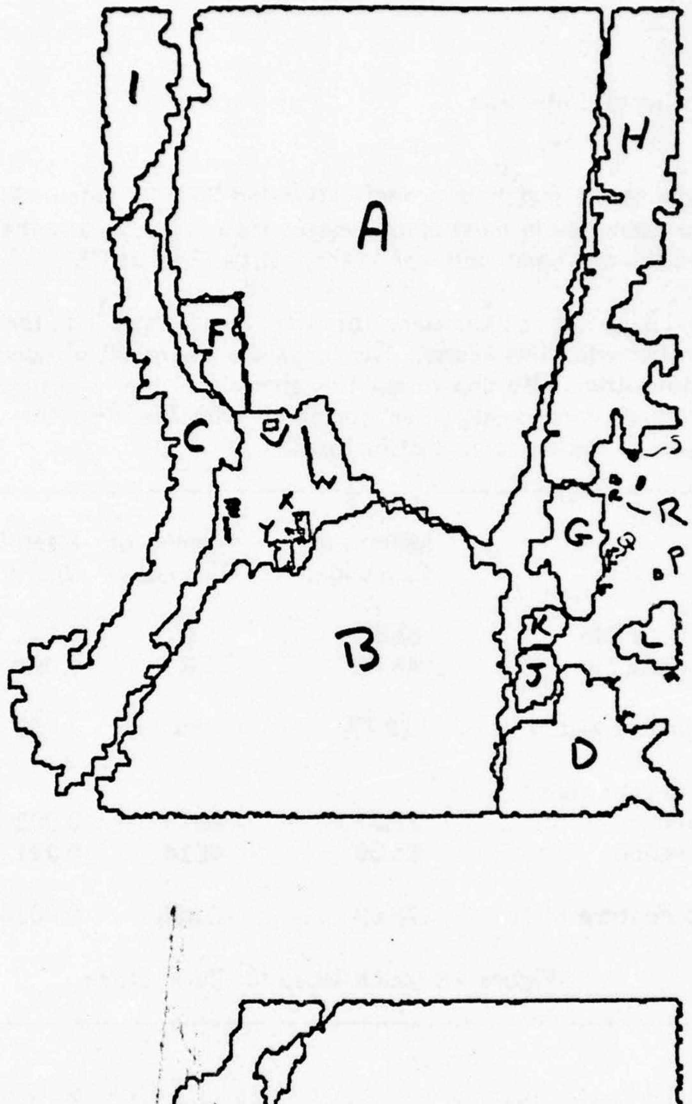


Figure 17 Rural Scene Match for 2 to 1

untextured regions all matched properly (labeled "A", "B", etc.). The bright regions near the center matched in most of the images ("Z", "Y", etc.), and the bright regions in the lower right (houses) matched well in the last two images ("S", "R", "Q", etc.).

Figure 18 gives a summary of the times required for several of the matching operations for this scene. The times are the result of combining four of the six matching operations into one total. This shows that the total time for the matching procedure itself is very small when compared with the time for any of the other operations (such as the extraction of features).

Operation	Millions of Operations	Number of Times Used	Mean Number of Operations
Read The Data File	55.53	8	6.941
Write The Data File	48.72	8	6.090
Region to Image Match	119.77	85	1.409
Region to Region Match			
Overhead	22.87	4514	0.005
Actual Match	96.59	4514	0.021
Feature to Feature Match	73.69	47333	0.0016

Figure 18 Match Times for Rural Scene

Figure 19 shows the full urban scene matching (only the bright regions were segmented). The correspondence between the regions marked "T" was used to limit the top of the pier area. The match of the regions marked "B" was used to indicate the bottom and left edge of the pier area. The results of the pier area matching were given in the previous section and indicated that 9 ships were located in the first image and 21 in the second. The actual count appears to be 7 (plus 3 or 4 much smaller ones) in the first image and 17 in the second. The errors are caused by two factors: some single objects are split into two regions, and a few non-ships are identified as ships for one reason or another. In the first image, one single ship was identified as a pair of ships because it was merged with one of the much smaller ships, and part of the ship was not segmented because it had a much different intensity and no micro-edges. These two factors caused the length to width ratio to be more similiar to the pair of ships than to a single ship. A small section of water was also identified as a ship, and a small piece of a ship at the top of the image was not segmented. The water region resembled a ship using length to width ratio, number of micro-edges, or intensity. Finally, a large block of water was indicated as a ship mostly because of the intensity and number of micro-edges. The size criterion should have eliminated it, but size was not used in this matching because it would introduce other errors in the smaller water and pier regions. In the second image, the errors are caused primarily by single objects being broken into two regions. Additionally, two pier sections are identified as ships and one small portion of a ship was not segmented. Two small (adjacent) ships were segmented as one region, but included some of the surrounding

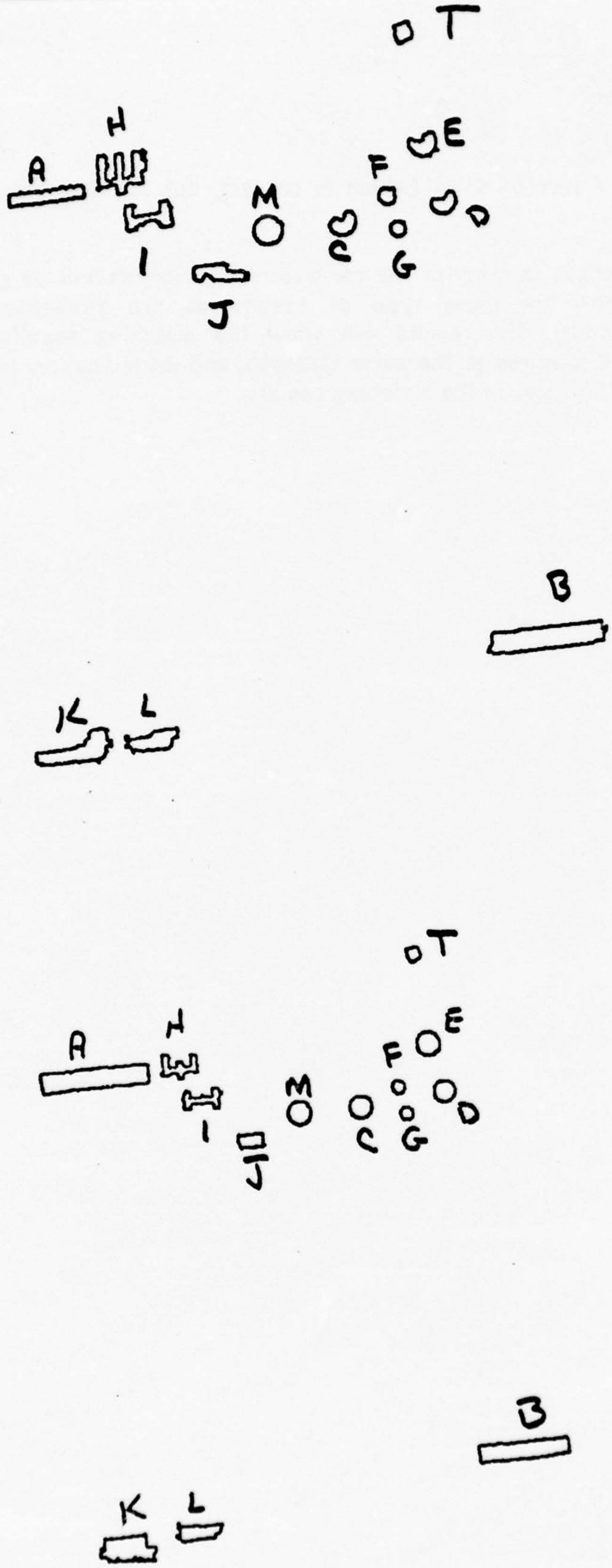


Figure 19 Urban Scene Match for Image 1 to 2

area and were indicated as a pair of ships (which is correct, but not for the right reasons).

Detailed results of changes in many of the corresponding regions will be given in Appendix 7. We will show the same type of results as are presented in Appendix 6 for matching results. The results will show the matching results for corresponding regions with all features at the same strength, and an indication of the changes which cause large differences in the matching results.

7 Summary and Conclusions

In this final chapter we will first offer a summary of the entire thesis and restate the primary results. We will then present the primary contributions of this thesis with a short description of each. The next section will be devoted to areas for future research.

7.1 Summary

This thesis describes research toward the development of a general image understanding system. Our system is directed toward the problem of the comparison of pairs of different images of the same scene to generate descriptions of the changes in the scene. Unlike earlier work in the change analysis area, we perform all the matching and change analysis at a symbolic level rather than a signal level. To facilitate this symbolic analysis over a wide variety of images, advances in several other areas of image analysis are also required. These areas are: segmentation techniques to generate the basic units used in the symbolic analysis, feature analysis to generate the symbolic description of the regions and image, use of knowledge to guide the segmentation and symbolic registration procedures, and lastly change analysis itself. We applied this procedure on several diverse scenes (house, cityscape, satellite images, aerial images, and radar images), each of which included a task description and a predefined set of knowledge elements, and have shown how several different tasks can be performed with a general change analysis system.

7.1.1 *Summary of the Tasks*

The scenes which we analyze (see Chapter 3 for a more complete description) are: a simple <u>house</u> scene, a <u>cityscape</u> scene, a <u>LANDSAT</u> (satellite) scene showing snow cover changes, a <u>SLR</u> (side looking radar) scene, an aerial <u>rural</u> scene, and an aerial <u>urban</u> or industrial scene. The first two scenes have three initial spectral inputs, the LANDSAT scene has four, and the other three have only one (intensity of radar signal or visible light). The tasks are: Perform simple symbolic registration for the house scene. Perform symbolic registration in a more textured scene with changes in the relative position of objects using the cityscape scene. Perform the analysis of a different spectral domain (radar) and symbolic registration with the SLR scene. Perform symbolic registration in the presence of rotations in the aerial rural scene. Perform symbolic registration and the analysis of the area of snow cover in the LANDSAT snow cover scene. And, finally, using knowledge guided segmentation, determine the change in the number of certain objects in the urban or industrial scene.

7.1.2 *Segmentation*

Chapter 4 presented the complete segmentation operations. We will give the high points here. Our work on segmentation is an extension of the histogram guided region splitting technique developed by Ohlander(1975). This method was originally developed for use on color images. Basically the procedure splits a region into subregions thresholding one of the spectral inputs. The threshold is selected by the analysis of the histograms of the values for all pixels in the region (one histogram for each spectral input). The threshold values are selected as the upper and lower bounds of the "best separated" peak which appears in the set of histograms. There

are two problems in the use of this technique for the segmentation of our set of images. First, the segmentation method is much too slow for processing a large set of images in a reasonably short time. Second, the segmentation technique was developed for multi-spectral images and could not be expected to work as well on the monochromatic images.

Planning: The first problem is solved by the introduction of "planning." By planning, we mean the generation of an approximation for the final segmentation using a reduced version of the image and the use of this approximation as a plan to more efficiently derive the true segmentation of the image. Ohlander gave a time of about ten hours for the segmentation of a color image with 0.5 million pixels (nine parameter for each pixel, each parameter represented by about eight bits). This time would be reduced to about five hours, if run now, because of modifications to many of the programs which he used (for example see Appendix 2 on modifications to the smoothing procedure). The use of planning reduces the total time to less than one half an hour (including the reduction time), or about one order of magnitude. There is also overhead involved in the manipulation of large images which is not reflected in these times. We present the segmentation times in hours rather than the number of operations which was used elsewhere in this thesis to enable the comparison with the times for Ohlander's segmentation. Both of these segmentation systems were run on the same computer system, so that the times are comparable.

Monochromatic Images: The segmentation of monochromatic images required additional alterations to the initial segmentation method. The original segmentation method was based on the hope that if one feature can not provide a reasonable split of the region, then, perhaps, another color parameter will. For example if two regions have the same intensity but are different colors, then the intensity parameter alone could not be used for segmentation, but another color parameter (possibly hue or Q) will. When the procedure is presented with only one spectral input, there is no other color parameter to turn to when there is only one peak in the histogram. The large monochromatic images also contained many small different objects which caused the histogram to have only one peak since the range of intensities for each region overlapped the ranges of intensity for other object.

We can introduce additional spectral-like features by the use of simple textural operators designed to show specific features such as homogeneous regions, or high contrast areas. We introduced a feature, the number of micro-edges in the reduction window, to indicate general homogeneous regions. A homogeneous region is one with few micro-edges so that these regions can be extracted by using a threshold of zero edges in the plan image. The points where the few edges occur will appear as small holes in the segmented region and are eliminated by the smoothing process. This threshold could not be applied directly to the initial micro-edge image (it is a binary image). The individual micro-edges would appear as small holes (a few points) in the thresholded image and would be swallowed up in the homogeneous region by the refining and extraction procedures. The smooth regions generated by the plan limit the area where this threshold is applied so that only a small number of edge points are swallowed up. The regions which are extracted are more sensitive to noise in the image, especially noise in one part of the image such as scratches. This feature is not generally useful for the extraction of exact regions, but proved useful for the extraction of general homogeneous regions.

Another textural measure is the excursion of values in the reduction window (maximum in the window minus minimum in the window). This measure is applied to the SLR scene to distinguish between the high contrast areas and the low contrast areas. This textural measure generated large general regions which correspond to the general textured areas. These were the only textural operators which were used in the segmentation of images.

Many other operators are possible, and for easy incorporation in a general segmentation method the operators should produce image like values for all points in the image. There are many possible textural operators, but we did not want to turn this thesis into an exploration of all possible texture operators. We do not want to judge the quality of other textural operators, but would be willing to incorporate others into this system.

We also used the computation of histograms for portions of the image rather than for the entire image. This was intended to approach a solution to the problem of many small similar regions. The use of partitions means that the number of separate objects which contribute to one histogram is reduced. If the partitioning of the image is extended as far as possible, at some point there will be only two distinct regions (or possibly one) to contribute to the histogram. At this point the threshold for segmentation would be obvious. Going to these extremes should not be necessary. We implemented a division of the image into only four or nine partitions.

<u>Evaluation of Segmentation:</u> When these alterations, textural measures and partial histograms, are coupled with the extended use of knowledge, it is possible to generate the regions necessary for performing the tasks for our monochromatic scenes. In a system which is meant to perform some task, the best measure for the quality of the segmentation is whether the segmentation is "good enough" to perform the task, such as are all the important regions extracted, and are these regions accurate enough to perform the task. Within the frame work of the tasks presented here, the segmentation process is sufficient. We are unable to analyze changes in small objects (i.e. only a few points) because this segmentation technique can not reliably separate these regions. Other segmentation problems arise in areas with many small irregular regions such as the area which includes the window of the house images. The small regions may be grouped into a few larger irregular regions, or no regions may be segmented because each individual region is too small to be accepted. The same problem would occur when the textural elements are large enough to appear in the plan image, but too small to be accepted by the plan generation.

Much of the segmentation procedure is now automated. The default criteria peak selection process is automated, so that the segmentation of the house and cityscape scenes was automatic. The special case peak selection procedures are not yet automated, such as forcing the procedure to find the bright peak, and the selection of homogeneous regions as regions with no edges. These selection criteria are used in special cases, but could be defined in a general pupose peak selection knowledge source, and the actual choice of criteria would be guided by outside knowledge or by the available data. When a high priority peak is composed of a few points (e.g. a few bright points) this peak will be selected every time since there is currently no way to force the selection procedure to ignore the high priority peak after discovering that it will segment no regions.

Most of the operations used in the segmentation procedures are amenable to implementation with special purpose processors. The use of special purpose processors to implement specific algorithms would be necessary if this (or any other image processing) were to be used on a large body of images, but such algorithms are currently tentative and experimental so that there is no need to expend the effort to build such processors at this time.

7.1.3 *Feature Extraction*

There are at least two very different techniques to give a symbolic representation of an image. One is a three-dimensional description of the objects in the scene such as representing all objects by a set of simple three-dimensional objects. This representation did not appear to be feasible to derive from general multiple views, and did not appear to be very useable for change analysis. We decided that the symbolic description would be composed of a set of regions which would be those generated by the segmentation procedure, and a set of features for each region describing various properties of the region. We group the features into classes similiar to those used by human beings performing the same sort of tasks. These feature classes include size, shape, color (including texture), location, and patterns. The exact feature measures were designed to capture various aspects of these feature classes. We computed the region size, absolute position of the center of mass, the position relative to other regions (above, below, etc.), adjacencies, average of color values or textural values, orientation, orientation independent length to width ratio, the fraction of the minimum bounding rectangle filled by the region, and the perimeter^2/area designed to indicate irregular regions. These are not all the feature measures which might be necessary for other tasks, and results should be more reliable when more features are available.

The methods for the computation of these features were not optimized and thus the computation times which are presented in Chapter 5 do not indicate the best attainable. The computation effort for some features is insignificant since these features are derived from other values (such as the length to width ratio, the orientation, fractional fill, perimeter^2/area). The expensive operations were the ones performed on all the points in the region, where most of the expense was in looking at the region points rather than computing feature values. The expensive features included: the color averages (mostly because they are used so often rather than being individually expensive), boundary computations (though it is less expensive than color averages since fewer points are accessed), orientation transformation computations (since they use the boundary computation), the initial color and texture transformations, and the relative position computations (which are expensive only because of the machine implementation). Like the segmentation operations, the expensive feature computations are amenable to implementation on special processors. The major descriptive feature which we did not study is the extensive use of textural measures.

7.1.4 *Symbolic Registration and Change Analysis*

The earlier systems for change analysis relied on correlation guided matching to locate corresponding point pairs and used the location differences of these point pairs either for transforming one image so that it is aligned with the other, or for depth analysis of stereo images. The aligned images are subtracted, producing a third

difference image. This difference image must then be analyzed to determine where the changes occurred, and what type of changes occurred. Special purpose systems have been built to perform these tasks, so that these apparently expensive operations are performed quickly. Change analysis systems which are intended to operate on uncontrolled image pairs (i.e. not stereo pairs) encounter several problems. The addition of more color parameters makes the problem more complex since the extra spectral inputs must be processed just like the initial input instead of simplifying the processing. Major changes in the point of view of the observer (especially in oblique views) will cause objects to change position with respect to each other and can cause inaccurate matches when those matches depend on intensity values in a neighborhood and are difficult (if possible) to account for in a global warping of the image. These systems used a "rubber sheet" warping so that points adjacent in one image are assumed to be adjacent in the other image. A new object in the scene can cause errors in matching, but such changes would usually be indicated as large differences in the difference image.

We present symbolic matching as an alternative matching technique to eliminate the problems encountered by earlier signal based change analysis methods. The addition of extra spectral inputs makes the segmentation processing easier and more reliable, and, if the desired regions are large enough, the use of planning means that the segmentation times will not be adversely affected by the addition of more inputs. Also, the addition of color parameters means that the matching procedures will have more features to use in the matching, this should also improve the reliability of the symbolic registration. Since the matching for one region does not necessarily depend on the intensity values in the image adjacent to the region being matched, the change in the relative position of objects should not reduce the chances of a correct match. We use many different features of the region including the adjacency and relative position relations, but the knowledge about the scene can specify that the relative position or adjacency relations will change: thus indicating that these features are not used for the symbolic registration. New objects are indicated by regions in the second image which had no corresponding region in the first image, and missing objects by regions which fail to match with any region. Finally, the change results produced by a signal based change analysis system are in the form of another image and must then be processed again to determine what changes have occurred. The symbolic change analysis system describes the changes as changes in the features of regions (or changes in the number of occurrences of an object). Thus there is no need for extensive processing of the resulting image to discover the kinds of changes, since these changes are given directly from the symbolic analysis.

<u>Symbolic Registration:</u> We developed a procedure which will determine a match rating for two regions in two different images. This rating procedure incorporates the differences between all available features of the regions. If the match is exact (e.g. matching a region with itself) then the rating will be zero, and as the match worsens, the rating decreases. The knowledge sources can indicate that certain features will change and thus should not be used in the matching procedure. For example, when the task description indicates that there are rotation differences between the two images, the matching procedure will not use the rotation dependent features such as the absolute position, the orientation, regions above, regions below, etc. Rather than eliminate the use of these features altogether, we introduce different strengths for features which should remain constant and features which will change. The strengths are selected so that a bad match in one feature that should remain

constant will have more impact than several bad matches in features which may change. This region to region match procedure is used in the symbolic registration procedure to find the best available match. To find the region in the second image which corresponds to a region in the first image, the symbolic registration procedure matches each possible pair of regions to find the best match. This best match is considered to be the corresponding region. Even if a region does not have a corresponding region in the other image, some region will be selected as the corresponding region. This region will be the most similiar region, but these two regions should have differences in features which should remain constant. Also, another region in the first image should correspond to the same region in the second image. This matching procedure has been applied to the six sets of images. We generated about a dozen sets of symbolic matching results (because we can match the second image to the first image in addition to matching the first image to the second image we can generate more sets of matching results than we have scenes).

Change Analysis: For some images we are given (through the image description) the fact that there is a scale change between images (as in the urban-industrial scene). The amount of the scale change is not given by the knowledge elements, but it can be computed from the size differences found in early matches. This scale change is used to adjust the size measures for regions in later matches. Since these is a scale difference between the two images, the absolute size and location features will change and can not be used as constant features in the matching operation. But with the use of the computed scale difference, the size feature can be used as if it is a constant feature. This use of the change results derived from the initial symbolic matching procedure can also be applied to the absolute location and orientation features, in addition to the size feature. These adjustments can apply only when the changes are uniform throughout the image, which is not the case when there are perspective changes as in oblique views. But such adjustments are possible to use in most aerial images.

Evaluation: Symbolic registration can be evaluated only by asking if all corresponding pairs are located correctly. The symbolic registration procedure performed very well when the regions were segmented consistently, but made some errors when the segmentations were very different. In all but the first two scenes (i.e. the SLR, rural, urban, and LANDSAT scenes), only a partial segmentation is generated so that it is not expected that all regions in one image will have a corresponding region in the second image. The lack of a corresponding region does not necessarily indicate that the region is missing, but may indicate that the region has different spectral characteristics. The house, cityscape, and pier subsection were all completely segmented and thus we can judge the matching for these scenes. The corresponding regions in the house scene were located very well since they are all large and well defined. The cityscape analysis made some errors matching part of a group of buildings. All the buildings in the group are the same color and the segmentation was not exactly the same in the two images, but it is very difficult to determine where one building ends and the next one begins. Probably these regions would be more meaningful if they were merged into a single silver-gray building. The cityscape also illustrated that symbolic matching could locate the corresponding regions in the presence of change in the relative position of objects. The pier subsection was used for the analysis of the changes in the number of "ship" regions in the two images. To perform this task we generated a pseudo-image containing a representative ship, pier, water, and shadow region. We then matched the two pier

area segmentations with this pseudo-image to determine which regions are "ships." Some of the "ships" were incorrectly segmented: they were broken into two regions, or only half of the region was segmented and the other half was merged with other regions (such as the piers or water). But the ship regions which were segmented were matched to a ship region. In the first subsection some errors occurred. The water regions were not as smooth and thus the average intensity and number of micro-edges in some of the water areas resembled the ship parameters more than the water parameters. In the second subimage errors were also caused by the matching of small parts of piers to ships because of the number of micro-edges in these pier regions. This was an attempt to extend the matching procedure into a rudimentary "recognition" procedure to compute the number of occurrences of a type of region feature.

The symbolic registration and change analysis processing is relatively fast when compared with all the other processing. This processing is best suited for implementation on general purpose computers rather than special purpose processors.

7.2 Contributions

In the previous section, the summary, we have mentioned several areas: segmentation, feature extraction, symbolic registration, and change analysis. In this section we will attempt to characterize specific contributions of this thesis in each of these areas.

7.2.1 Segmentation

We adopted an existing segmentation procedure which had already proved to be useable on many different types of scenes. This method had two major problems: it was too slow and it was designed for multi-spectral images. We introduce the use of planning to make the segmentation faster, which resulted in a speed-up factor of ten. To use the procedure on multi-spectral images, we incorporated two simple textural operators into the planning system and added the use of knowledge in the peak selection processing. The use of knowledge was necessary to force the peak selection to find a specific peak rather than the "best" peak. These modifications made it possible to acquire the partial segmentations required for our tasks. It is difficult to evaluate the quality of segmentation of natural images, other than to ask if the segmentation produces the regions necessary for the performance of the current task. It is rather difficult to accurately hand segment the regions in an image, so that this is not practical to use as a comparison. In this respect, the segmentation is adequate, but not perfect.

7.2.2 Features

The features which we have used are all taken from the literature. The exact implementation of some of the features may be novel, but this is not important. The importance rests in the use of a feature based description system in a change analysis system. This system is not dependent on the particular features which are implemented; the incorporation of new features is straight forward for new researchers.

7.2.3 *Symbolic Registration*

We developed a feature based region comparison procedure to determine "how well" two regions matched and used this procedure to locate the corresponding region pairs for the segmented regions in a diverse set of scenes. Symbolic matching is designed to eliminate many of the problems encountered by signal-based matching. Changes in the relative position of regions do not adversely affect the symbolic matching, since the location features are only a small subset of the features used for matching. The addition of more spectral inputs simplifies the matching procedure rather than complicates it.

We not only matched regions, but we used certain differences between corresponding regions for later region matches. The size and location adjustment factors were necessary for some of the symbolic registration operations using the urban-industrial scene. And, the location differences were required for most of the matches in the SLR scene.

7.2.4 *Change analysis*

Two examples of change analysis are presented. The first is just the difference in the size of corresponding regions, for example the difference in the snow cover for a particular area. The second example shows that symbolic analysis can produce results as a symbolic description rather than as an image which must be further processed. For example, the results are simply the location and number of "ship" regions in two images.

7.2.5 *Summary*

We intended this work as a general change analysis system rather than as a system for the solution of a specific problem. However, if a system is intended to be the solution of a specific problem using a well defined set of data, then it is clear that the system can include many special purpose techniques. This work is not proposing symbolic analysis as the solution of every "simple" well defined problem.

7.3 Future Research

This section will present several areas of future research in symbolic analysis of changes and other possible applications of this work.

7.3.1 *Segmentation*

The evaluation of a segmentation of a natural scene is difficult. There is no easy way to generate an accurate hand segmentation (in a reasonable time), so that the hand segmentation could be compared with the machine segmentation. The best current evaluation method is to determine if the segmentation is sufficient for the task, but this will become more difficult as tasks become less well defined.

The segmentation method produced some "ragged" regions, and, with the use of texture operators, some very general regions. Regions of this type should be refined

so that they correspond more closely with the real world objects. After the region in the plan image is expanded to the full size image, the refinement could be the application of the segmentation procedure to eliminate the "tails" on the histogram peaks, rather than the application of a single threshold. This should produce cleaner region boundaries.

Sometimes a single object is broken into two separate regions. Perhaps a feature based region merging system could be applied to the segmented image to merge similiar adjacent regions. There should be differences between the adjacent regions (or else they would not have been split apart), but the regions could be joined if the merging processor could "explain" the differences. Explainations would include shadows, or image flaws such as scratches.

We only touched briefly on the use of texture in segmentation and feature analysis. Texture is very important for human analysis of scenes and can lend much to change analysis. There is already a large body of work on texture analysis for classification of regions, but we can not yet judge the usefulness of this work with respect to its use in a general change analysis system.

7.3.2 Symbolic Representations

We presented only a few features, but many more are needed of effective analysis of a larger set of scenes. Many more features are available from texture analysis, but there are also other size features (such as the largest, the horizontal extent, etc.), location features (such as location of extremes), and shape features which could be added.

7.3.3 Symbolic Change Analysis

Overall work is needed in the incorporation of more outside knowledge in the change analysis procedure. These additions include the more automatic selection of the features based on the knowledge of what can change based on the the observer induced changes.

Change analysis would also be improved by advancements in the segmentation procedure. If the change analysis procedures are presented with perfect segmentations then they should work correctly.

This change analysis system is not limited to the comparison of two images. With the addition of more features we could incorporate the computation of changes through a sequence of images, by using the change results for the first to the second to generate expected changes for the second to the third. The processing would then confirm the changes and generate any new changes.

The system could also be used for the comparison of an image with a feature based representation of a world model, such as a map, to update the world model.

7.4 Conclusion

This thesis has been a preliminary effort in the use of symbolic techniques for change analysis. The results of this thesis indicate that symbolic analysis offers the best chance for the analysis of a large class of images by a general change analysis system, but there still remains much work before a truely general complete system is built. As systems are built for more detailed results, they will need to incorporate more and more task specific knowledge in the segmentation, both in the choice of features, and also in the matching procedures. Symbolic techniques, of the type described here, offer the best choice for the incorporation of this knowledge and success in computer change analysis of natural scenes.

8 Bibliography

Agin, G. J. (1972), "Representation and Description of Curved Objects," AIM-173, Thesis (Comp. Sci.), Stanford University, Stanford, CA, October, 1972.

Akin, O., and R. Reddy (1976), "Knowledge Acquisition for Image Understanding Research," Computer Graphics and Image Processing, to appear, 1976.

Allen, G. R., L. O. Bonrud, J. J. Cosgrove, and R. M. Stone (1973), "The Design and Use of Special Purpose Processors for the Machine Processing of Remotely Sensed Data," IEEE Symposium on Machine Processing of Remotely Sensed Data, Purdue University, October, 1973.

Badler, N. I. (1975), "Temporal Scene Analysis: Conceptual Descriptions of Object Movements," Technical Report Number 80, Department of Computer Science, University of Toronto, February, 1975.

Bajcsy, R. (1972), "Computer Identification of Textured Visual Scenes," AIM-180, Thesis (Comp. Sci.), Stanford University, Stanford, CA, 1972.

Barrow, H. G., and R. J. Popplestone (1971), "Relational Descriptions in Picture Processing," Machine Intelligence 6, Meltzer, B., and Michie, D. (eds.), University Press, Edinburgh, 1970, pp. 377-396.

Blackmer, R. H., R. O. Duda, and R. Reboh (1973), "Application of Pattern Recognition Techniques to Digitized Weather Radar Data," Final report to NASA, Stanford Research Institute, Menlo Park, CA, May, 1973.

Brice, C. R., C. L. Fennema (1970), "Scene Analysis Using Regions," AI Journal 1, Fall 1970, pp. 205-226.

Chow, C. K., and T. Kaneko (1970), "Boundary Detection of Radiographic Images by a Threshold Method," IBM Research Report RC-3203, 1970.

Clowes, M. B. (1971), "On Seeing Things," AI Journal, Spring 1971.

Duda, R. O., and R. H. Blackmer (1972), "Application of Pattern Recognition Techniques to Digitized Weather Radar Data," Final report to NASA, Stanford Research Institute, Menlo Park, CA, March, 1972.

Duda, R. O., and P. E. Hart (1973), Pattern Classification and Scene Analysis. New York: Wiley-Interscience, 1973.

Falk, G. (1970), "Scene Analysis Based on Imperfect Data," in Proc. of 2-IJCAI, London 1971.

Gibson, J. J. (1950), The Perception of the Visual World, Boston: Houghton Mifflin Co., 1950.

Grape, G. R. (1973), "Model Based (Intermediate Level) Computer Vision," AIM-201, Thesis (Comp. Sci.), Stanford University, Stanford, CA, May 1973.

Gupta, J. N., and P. A. Wintz (1974), "Multi-Image Modelling," TR-EE 74-24, School of Electrical Engineering, Purdue University, West Lafayette, Ind, September, 1974.

Guzman-Arena, A. (1968), "Computer Recognition of Three-dimensional Objects in a Visual Scene," MAC-TR-59, Thesis (EE), MIT, Project MAC, Cambridge, MA, December, 1968.

Haber, R. N. and M. Hershenson (1973), The Psychology of Visual Perception, New York: Holt, Rinehart, and Winston, 1973.

Hanna, J. (1974), "Computer Matching of Areas in Stereo Images," AIM-239, Thesis (Comp. Sci.), Stanford University, Stanford, CA, July 1974.

Hanson, A. R., and E. M. Riseman (1974), "Preprocessing Cones: A Computational Structure for Scene Analysis," Technical Report 74C-7, Computer and Information Science, University of Massachusetts, Amherst, MA, September, 1974.

Hanson, A. R., and E. M. Riseman (1975), "The Design of a Semantically Directed Vision Processor," Technical Report 75C-1, Computer and Information Science, University of Massachusetts, Amherst, MA, February, 1975.

Hanson, A. R., E. M. Riseman, and P. Nagin (1975), "Region Growing in Textured Outdoor Scenes," Technical Report 75C-3, Computer and Information Science, University of Massachusetts, Amherst, MA, February, 1975.

Haralick, R. M., and D. E. Anderson, "Texture-Tone Study with Applications to Digital Images," ETL-CR-71-14, Center for Research, University of Kansas, Lawrence, Kansas, November, 1971.

Hueckel, M. (1973), "A Local Visual operator Which Recognizes Edges and Lines," JACM 20 (1973), pp. 634-647.

Huffman D. A. (1969), "Logical Analysis of Pictures of Polyhedra," AI Group Tech. Note No. 6, SRI project 74014, SRI, Menlo Park, Ca, May, 1969.

Hunt, R. W. G. (1967), Reproduction of Colour, Wiley and Sons, New York, 1967.

Kelly, M. D. (1970), "Visual Identification of People by Computer," AIM-130, Thesis (Comp. Sci.), Stanford University, Stanford, CA, July, 1970.

Kender, J. (1976), "Saturation, Hue, and Normalized Color: Calculation, Digitization Effects, and Use", Technical Report, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1976.

Levine, M. D., D. A. O'Handley, and G. M. Yagi (1973), "Computer Determination of Depth Maps," Computer Graphics and Image Processing, (1973), 2, pp. 131-150.

Lillestrand, R. L. (1972), "Techniques of Change Detection", IEEE Transactions on Computers, Vol. C-21, No. 7, July, 1972.

Miller, G. A. (1956), "The Magic Number Seven, Plus or Minus Two: Some Limits On Our

Capacity for Processing Information," in Contemporary Theory and Research in Visual Perception, Haber, R. N. ed., New York: Holt, Rinehart and Winston, Inc., 1968, pp. 187-202.

Nevatia, R., and T. Binford (1973), "Structural Description of Complex Objects," 3-IJCAI, Stanford, Ca, pp. 641-7, 1973.

Ohlander, R. (1975), "Analysis of Natural Scenes," Thesis (Comp. Sci.), Carnegie-Mellon University,, Pittsburgh, PA, June 1975.

Prewitt, J. M. S. (1970), "Object Enhancement and Extraction," in Picture Processing and Psychopictorics, Lipkin and Rosenfeld eds., New York: Academic Press, 1970, pp. 75-149.

Quam, L. H. (1971), "Computer Comparison of Pictures," AIM-144, Thesis (Comp. Sci.), Stanford University, Stanford, CA, May, 1971.

Roberts, L. G. (1963), "Machine Perception of Three-Dimensional Solids," in Optical and Electro Optical Information Processing, Tippett, J. T., et. al., eds., Cambridge: MIT Press, 1965, pp. 159-97.

Rosenfeld, A. , Picture Processing by Computer, Academic Press: New York, 1969.

Sakai, T., M. Nagao, T. Kanade (1972), "Computer Analysis and Classification of Photographs of Human Faces," Kyoto University Report, Kyoto, Japan, 1972.

Shirai, Y. (1972), "A Heterarchical Program for Recognition of Polyhedra," AI Memo-263, MIT AI Laboratory, Cambridge, MA, June, 1972.

Tenenbaum, J. M., T. D. Garvey, S. Weyl, H. C. Wolf (1974), "An Interactive Facility for Scene Analysis Research," Artificial Intelligence Center Technical Note 87, Stanford Research Institute, Menlo Park, CA, January, 1974.

Tenenbaum, J. M., H. G. Barrow, and S. A. Weyl (1976), "Research in Interactive Scene Analysis," Final report to NASA, Stanford Research Institute, Menlo Park, CA, 1976.

Tomita, F., M. Yachida, S. Tsuji (1973), "Detection of Homogeneous Regions by Structural Analysis," in Proc. 3-IJCAI, Stanford University, Stanford, CA, 1973, pp. 564-71.

Underwood, S. A. and C. L. Coates (1975), "Visual Learning from Multiple Views," IEEE Trans. on Computers, C-24, No. 6, June, 1975, pp. 651-661.

VanLehn, K. A. (1973), "SAIL User Manual," AIM-204, Stanford Artificial Intelligence Laboratory, Stanford, CA, July, 1973.

Waltz, D. L. (1972), "Generating Semantic Descriptions from Drawings of Scenes with Shadows," AI-TR-271, Thesis (EE), MIT AI Laboratory, Cambridge, MA, November, 1972.

Winston, P. (1970), "Learning Structural Descriptions from Examples," MAC-TR76, Thesis (EE), MIT Project MAC, Cambridge, MA, September, 1970.

Yakimovsky, Y. (1973), "Scene Analysis Using a Semantic Base for Region Growing," AIM-209, Thesis (Comp. Sci.), Stanford University, Stanford, CA, July, 1973.

9 Glossary

Feature: Symbolic descriptor of an object or region.

Histogram: The plot of the number of occurrences of all possible values; a frequency distribution.

Image: Digital representation of a scene in one or more spectral bands.

Mask: Binary image showing a region which represents an object or objects.

Minimum Bounding Rectangle: (MBR) The rectangular subwindow that minimally contains the regions.

Object: A distinct part of the scene.

Pixel: Picture element, also called pel or point.

Plan: The form of something to be done.

Region: A group of points in an image, corresponding to an object, part of an object, or a collection of objects in the scene.

Scene: The "real" world, represented by one or more digital images.

Segmentation: Division of an image into discrete regions with similiar properties.

Smooth: Low pass filter operation.

Template: Mask used to describe expected shape of an object.

Threshold: A cutoff value; lower and upper bounds; level slicing.

Window: Neighborhood of a certain size.

Appendix A Normalized Correlation Computation

We have mentioned the expense involved in the computation of correlation values. The correlation match rating between two points is the normalized cross correlation of the points in the neighborhoods of the two points. The normalized cross correlation of two sets of N elements (X and Y) is defined as:

$$\frac{\mathrm{cov}(X,Y)}{\sigma_X * \sigma_Y}$$

$$= \frac{\sum(X_i Y_i)/N - \bar{X} * \bar{Y}}{\mathrm{SQRT}((\sum(X_i^2)/N - \bar{X}^2) * (\sum(Y_i^2)/N - \bar{Y}^2))}$$

$$= \frac{\sum(X_i Y_i) - N * \bar{X} * \bar{Y})}{\mathrm{SQRT}((\sum(X_i^2) - N * \bar{X}^2) * (\sum(Y_i^2) - N * \bar{Y}^2))}$$

Program segment for a general cross-correlation operation:

```
for i←1 step 1 until N do
    begin
    ytot←ytot+y[i];
    xtot←xtot+x[i];
    sumy2←sumy2+y[i]*y[i];
    sumx2←sumx2+x[i]*x[i];
    sump←sump+x[i]*y[i];
    end;
corval←(sump-xtot*ytot*N)/SQRT((sumx2-xtot*xtot*N)*(sumy2-ytot*ytot*N));
```

Note: the summation for X alone can be done once for each search, but this is still very expensive to apply to many points in the image. This computation produces the correlation match for two points. To find the best match in a second image for a point in the first image, this procedure must be applied between the point in the first image (i.e. X) and many points in the second image (i.e. Y).

Appendix B Use of Smoothing

In the segmentation chapter (Chapter 4) we mentioned the use of "smoothing" to enlarge a binary mask and for the elimination of holes in regions by enlarging and shrinking the mask. This Appendix presents the general method and describes a procedure to reduce the computation effort when the smooth window size is greater than two by two.

Generally speaking, smoothing is considered to be replacing a point in an image (or anywhere else) with a value which depends on the surrounding values; for example an average. The following program segment illustrates a possible implementation of a smoothing operator which averages the "window" by "window" neighborhood of points in the image (array) PIC and stores the results in NewPIC:

```
for i←-window%2 step 1 until window%2 do
    for j←-window%2 step 1 until window%2 do
        value←value+PIC[i+pi,j+pj];
NewPIC[pi,pj]←(value/window^2)+0.5;
```

In this program segment % denotes integer division. This computes the average of the values in the window around PIC[pi,pj] and stores this result in NewPIC[pi,pj]. This implementation is straight forward, and shows what is being done, but is extremely slow. The "0.5" added in the final line of the program is used to round the result since this program is in SAIL on a PDP-10, and storing the real (floating point) result of the division into an integer value truncates the number to the integer part of the number. A final comment on this program is that it is valid only for odd window sizes, and for points away from the edge of the image.

Observe: for binary images, if the "0.5" rounding factor is replaced by a variable, then the smooth operator can be used to set a pixel to "1" depending on the number of "1" pixels in its neighborhood. For example, if the rounding factor is $1/\text{window}^2$, then a point will be set to "1" only when there is at most one pixel in the neighborhood with a value of "0". When it is $(\text{window}^2-1)/\text{window}^2$, then pixels will be set to "1" if there are one or more pixels with a value of "1" in the neighborhood.

This procedure is a robust one. It is a general purpose smoothing (low pass filter) program when the rounding factor is 0.5, and is usable for special processing of binary images.

The above implementation is several orders of magnitude too slow for general use, but there is a straight foreword procedure to obtain a substantial speed-up. Figure 1 illustrates what the variables in the description correspond to. "A" is the point above the current point ("B"). The neighborhood of "A" includes the large square around "A" and "B" plus "TOP". The neighborhood of "B" is the large square plus "Bottom". "TR" and "BR" are included in "TOP" and "Bottom" respectively. Thus:

$$\text{Sum for B} = \text{Sum for A} - \text{TOP} + \text{Bottom}$$

and

$$(\text{TOP} - \text{Bottom}) = (\text{Last\_TOP} - \text{Last\_Bottom}) + (\text{BR} - \text{BL}) - (\text{TR} - \text{TL})$$

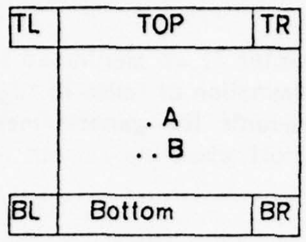


Figure 1 The Faster Smooth Operation

Where Last_TOP and Last_Bottom were TOP and Bottom computed for the previous point. Therefore, the two nested loops in the above program segment can be replaced by two simple statements:

```
adjust←lastadjust - BL + BR + TL - TR;
value←Value_for_A + adjust;
```

This computation method is not valid for the first row and column that are computed. Further the sum for one complete row must be stored. No one would consider using the first program segment for smoothing (at least not more than once), but would usually only compute the sum for the column of the window that is added relative to the window for the last point.

This same procedure is also used for the Ohlander busy-nonbusy computation (which computes the sum of the number of points set to "1" in the neighborhood). Here do not divide by the window size, or add any rounding factor (i.e. divide by one and add zero).

Appendix C Timing Information

In earlier chapters we have mentioned the cost of various image processing operations in terms of the number of basic operations. This appendix gives the number of operations necessary for an "ideal" implementation of the processing step. The columns give the cost in terms of the number of loops whose overhead is included, the number of basic operations (e.g. an arithmetic operation), and the number of references to primary and secondary memory to reference and save the picture points.

Operation	Operations per pixel loop	OPS	$M_{Primary}$ references	$M_{Secondary}$ references
YIQ DHS (Color Transforms)				
total	1	51	9	9
Y		6	1	1
I		6	1	1
Q		6	1	1
Density		5	1	1
Hue		12	1	1
Sat		6	1	1
Other	1	10	3	3
Texture Computation				
Zero Crossings	1	45	1	1
Edge Operator	1	40	1	1
Normalize	1	5	2	2
Translation	1	2	2	2
Segmentation operations:				
Histogram	1	6	1	1
With Mask	1	8	2	2
Threshold	1	4	1	1
With Mask	1	6	2	2
Smoothing	1	13	6	2
Select Connected Regions	1	20	3	3

Note: When a mask on a portion of the picture is used (generally of a size of less than one half of the entire picture) the figures are for points that are "1" in the mask. Points that are "0" require fewer operations. The times given here are for ideal implementations on a general purpose computer and do not necessarily reflect the times for a particular machine. Some general purpose machines will require many more instructions, while special purpose image processing machines will perform one of these operations on a set of pixels as a single instruction.

Appendix D Border Follow Routine

We have mentioned the use of a border following procedure for the computation of the neighbor relation. This procedure is also used for the extraction of connected regions, and the generation of outline drawings of the segmented regions. This procedure will follow the outline of an eight connected region.

```
internal integer procedure border(integer start_i,start_j,input_picture,outline_buffer;
      reference integer imin,imax,jmin,jmax; integer region_number);
   begin "borderfollow"
   integer array neighbors[0:7];
   integer regnum,m,next,i,j,start,temp,i_index,j_index,numpt,offset_i,offset_j;
   i←start_i; j←start_j;                                              ! initial starting values;
   offset_i←isubst(input_picture)-isubst(outline_buffer);         ! offsets between the picture buffer;
   offset_j←jsubst(input_picture)-jsubst(outline_buffer);                 ! and the outline buffer;
   start←numpt←0;
   for m←0 thru 7 do
      begin "load1"
      i_index←i+(case m of (0,-1,-1,-1,0,1,1,1));
      j_index←j+(case m of (1,1,0,-1,-1,-1,0,1));
      if 1 ≤ i_index ≤ MaximumI and 1 ≤ j_index ≤ MaximumJ
            then neighbors[m]←getpnt(i_index,j_index,input_picture)
            else neighbors[m]←0;
      end "load1";
   for next←0 step 1 while next<8 ∧ neighbors[next] do;                 ! find the first neighbor;
   if next>7 then return(0);                              ! bad starting point - no "1" pixels;
   for next←next+1 thru 7 do if neighbors[next] then done;
   if next=8 ∧ ¬ neighbors[next←0] then return(-1);                        ! a single point region;
   while true do
      begin "loop"
      temp←(next+start) mod 8;
      i←i+(case temp of (0,-1,-1,-1,0,1,1,1));                 ! go to the next point in the boundary;
      j←j+(case temp of (1,1,0,-1,-1,-1,0,1));
! here is where the various actions are added, such as calculations based on outline coordinates;
      putpnt(i+offset_i,j+offset_j,region_number,outline_buffer);
      numpt←numpt+1;                                           ! number of points along the boundary;
      if i<imin then imin←i else if i>imax then imax←i;              ! save the limits of the outline;
      if j<jmin then jmin←j else if j>jmax then jmax←j;
      if i=start_i and j=start_j then return(numpt);                   ! finished with the perimeter;
      start←if (temp LAND 1)
            then if (temp←temp-3)<0 then 8+temp else temp
            else if (temp←temp-2)<0 then 8+temp else temp;
      next←0;
      for m←0 thru 7 do                                            ! select the next boundary point;;
         begin
         temp←(m+start) mod 8;
         i_index←i+(case temp of (0,-1,-1,-1,0,1,1,1));
         j_index←j+(case temp of (1,1,0,-1,-1,-1,0,1));
         if 1 ≤ i_index ≤ MaximumI and 1 ≤ j_index ≤ MaximumJ
            then if (neighbors[m]←getpnt(i_index,j_index,input_picture)) ∧ next=0 then next←m
                                                          else
            else neighbors[m]←0;
         end;
      end "loop";
   end "borderfollow";
```

Appendix E Processing Example

This appendix presents a description of all the processing which is required to generate the final results for the change analysis task of the pier subsection of the urban-industrial scene.

To review: These images are given in Figure 3.13 for the first image and Figure 3.14 for the second image. The images have about four million pixels of eight bits each, containing the intensity value only. The images were taken at different times (weeks or months apart) and at a different time of day causing changes in objects and in shadows. The first image is at a larger scale than the second and the top left corners of the two images do not correspond to the same point, so that there are observer induced changes in the size and position of objects. The images have been aligned so that there is no rotation difference between them. The task comes in two parts: the symbolic registration of bright regions in the full image, and the detection of the change in the number of ships in the pier area.

The processing steps are:

1. Segmentation of the bright regions in the two images
2. Extraction of features
3. Symbolic registration of a few regions
4. Symbolic registration of all regions using changes derived from step 3
5. Selection of the regions in image 1 to determine the pier subsection, and the extraction of this area in both images
6. Generation of the textural operators for the pier subsection
7. Refinement of the pier subsections with regions from a first segmentation
8. Complete segmentation of the remaining pier subsection
9. Extraction of features for the segmented images
10. Selection of representative regions for the computation of the number of occurrences feature
11. Matching of both images with the pseudo image
12. Counting the results

These steps will be explained in the following subsections.

E.1 Segment the Bright Regions

The task description and knowledge indicated that the bright regions are to be used in the later processing so that these regions must be extracted. The details of this segmentation were given in the subsection on partitioning in Chapter 4. Because the images contained so many similiar regions, the bright peak is hidden in the histogram of the full size image. Partitioning the image into nine equal subimages causes a bright peak to appear in several of the subimages. The threshold of 190 to 255 was used for the first image and 220 to 255 for the second. The current peak selection program can not be forced to take a specific peak so the selection of the peak is done manually, but the bounds on the selected peak are given automatically. These segmentations were generated with the plan using a reduction factor of eight, and the expanded results are given in Figures 1 and 2. Some of the regions in these segmentations are very irregular in shape, these were probably the regions which filled the valley between the peak for the bright regions and the average regions.

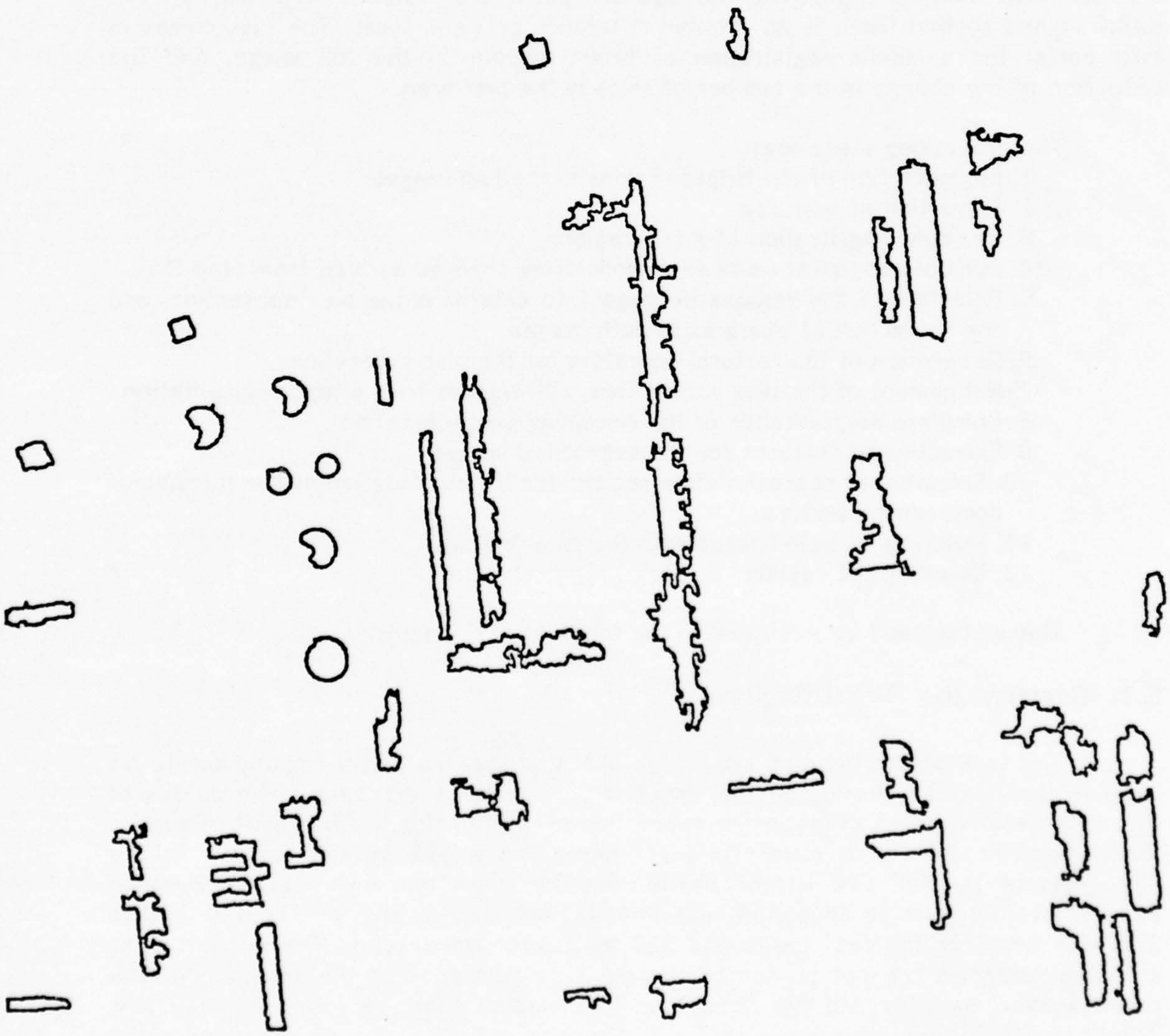


Figure 1 Urban Scene Image 1 Segmentation

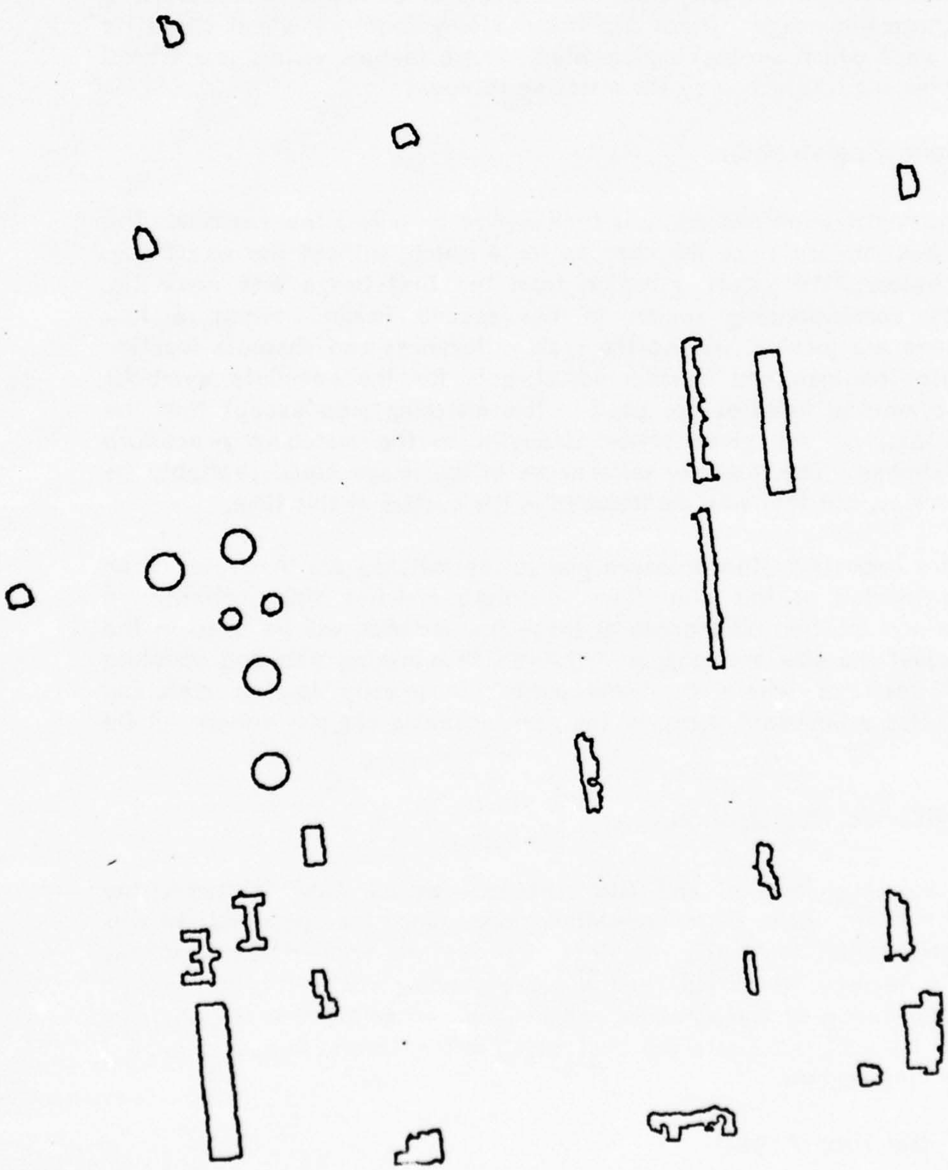


Figure 2 Urban Scene Image 2 Segmentation

Information files are generated automatically by the segmentation operations, and are used throughout the processing.

E.2 Extract the Features

The features of every segmented region are extracted by the same program which will be used for matching. When the program is used for matching it will use the previously computed values and will not recompute the feature values. The feature value for all possible features are computed. An example of an impossible feature is red for the monochromatic image. These are not the only features which could be computed; just the ones which we had implemented. These feature values are stored in the information files for future use by the matching routines.

E.3 Initial Symbolic Registration

The symbolic registration procedure is first applied on only a few regions. The larger regions are best to use since the chances for a match, without the exact size and location, are better. We select a region from the first image and have the procedure find its corresponding region in the second image. When a few corresponding regions are located, we use the scale differences and absolute location changes to calculate the size and location adjustments for the complete symbolic matching. All the computed features are used in this matching step except that the location and size features are given lower strengths in the matching procedure because they will change. The intensity differences in the image could probably be handled in a similiar way, but that was not included in the system at this time.

This is a very important step to insure that future matches are accurate. If an improper match is located at this step then all future matches will probably be incorrect. The size and location differences of these few matches will be used in the later matches to adjust the size and location measures, thus making size and absolute location "constant" features where they will contribute greatly to the matching operation. Without this adjustment, many of the corresponding regions would not be correctly located.

E.4 Complete Symbolic Registration

Using the adjusted location and size measures as constant features, the symbolic registration is applied to the segmented regions. Since the segmentation was only a partial segmentation (only the brightest regions), we applied the symbolic registration only to regions which did have a corresponding region in the second image. For each application of the symbolic registration, we selected a region from image one and had the program locate the best match in the second image. Figure 3 gives the corresponding regions.

E.5 Selection of the Pier Area

Since the task description specified the analysis of a subportion of the image, we implemented a knowledge source to extract a subsection of two images. The bounds of the subsection in the first image are given in terms of the regions which specify the extremes. The second subsection is derived from the first by using the

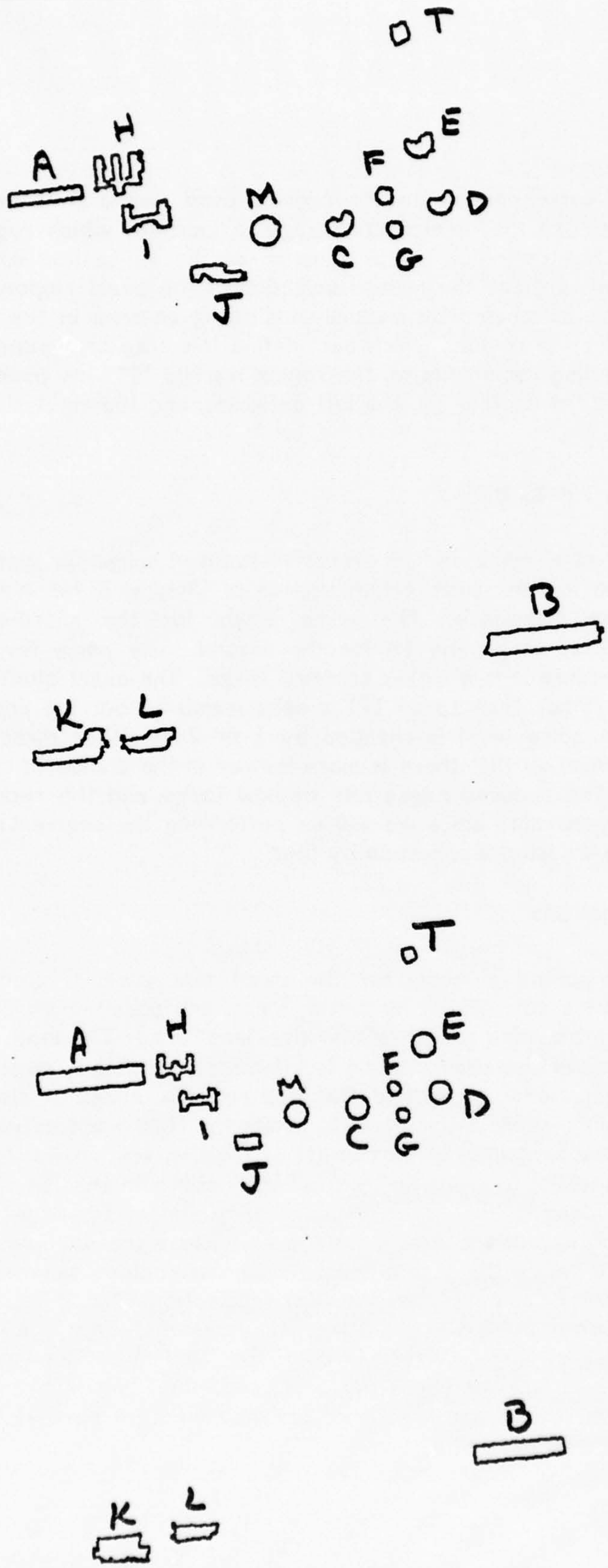


Figure 3 Urban Area matching Results

regions in the second image corresponding to the regions used in the first image. Therefore, we must select (i.e. use the outside knowledge to indicate) which regions are to limit the subsection. The definition of the location of the subsection of the image which is desired is used to guide the manual selection of the exact regions (in the first image). This could be automated by manually indicating an area in the first image and automatically locating the regions which best define this area or a superset of the area. In the corresponding region figure, the region marked "T" was used for the top, "B" was used for both the bottom and the left extremes, and the right side is the right edge of the image.

E.6 Compute the Textural Features

We computed a micro-edge image and an excursion reduced image for both of the pier subsections. See the texture computation section of Chapter 5 for a more complete discussion of these operators. The noise levels for the micro-edge computation are 15 for the first image and 18 for the second. The noise level is higher in the second image because it is a higher contrast image. The exact choice of the noise level (i.e. using 15 rather than 13 or 17) is done manually, but the results are changed very little if the noise level is changed by 1 or 2 in either direction. These images are 8 bits per pixel so that there is more leeway in the choice of noise levels than in 6 bit images. The reduced edges per window image and the reduced intensity image must also be generated since we will be performing the segmentation with planning. In this example we used a reduction by four.

E.7 Refinement of Pier Sections

We derived and implemented a model for the exact pier area: it contains regions representing water, piers surrounded by water, ships, and possibly shadows. The water, piers, and ships are bounded on the left by the "land" area. The land area is removed from the image before the segmentation is attempted, so that it does not interfere with the segmentation. The removal of the land requires a set of clearly segmented regions such as water, shadows, or piers to locate the right most extremes of the land. This is done by the following: given this list of regions, set the left limits of the pier area as a line connecting the left limits of all the regions in the list. This line is really a collection of straight line segments connecting the extremes of the regions from the top to the bottom of the image. In the first image the shadows (of the ships and piers) were used and in the second image the water regions were used. The final selection of the regions is manual, after the first segmentation has generated some regions, and the extraction is automatic given the list of regions. This is a very important step in the processing of these images. Much of the "land" area "looks like" a "ship" region in terms of the number of edges per area, and intensity which are to be used for segmentation. The "ships" and the "land" are also adjacent so that they would tend to be segmented as one region.

E.8 Complete Segmentation

Perform the segmentation process on the two remaining pier subsections. The segmentation of the second image is straight forward: ships contain lots of micro-edges, and the piers do not. There is also an intensity difference between the two. The first image presents some problems since the "water" in the lower part of the

image also contains many micro-edges (due to waves), and is thus a different average intensity than expected. This causes the water to sometimes blend in with the ships. But, a straight forward application of the segmentation procedure generates all the regions. Some "ships" are split into two regions and pairs of "ships" are segmented as one region, but the segmentation is good enough to use. These final segmentations are given in Figures 4 and 5.

E.9 Extract the Features

The feature extraction procedure is the same as used in the bright region feature extractions. Except, in this case neighbors exist, and they did not in the original segmentation. Also, since the micro-edge image is used, the edges per window can be computed.

E.10 Pseudo Image

The task statement indicates that we must compute the number of occurrences of a certain type of object. We have no recognition procedure available to classify the regions, so we generated a pseudo image to use in the matching procedure. Figure 6 shows the regions which are to be used. Some of these regions are from the first image and some are from the second so that the location of the individual regions is not important. This pseudo image is a model of which objects can appear in the scene. This model is constructed with a representative region for all the possible types of regions: "pier,", "water," "shadow," and "ship." Since the "ship" regions covered a wide range of values, we also included a "ship" pair region, and a long "ship." The pseudo image was constructed from regions segmented in the two images, by hand with no analysis used to pick the "most" representative region.

E.11 Match the Images

All the regions in the two images are matched with the pseudo image. Some features are impossible to use, such as the absolute position, and relative positions. The remaining features include: orientation, height to width ratio, color (including the number of micro-edges in a window), neighbors, size, fractional fill, and perimeter^2/area. The matching results are indicated in Figures 7 and 8, "S" means a ship, "S2" is two ships, "W" is water, "P" is pier, and "H" is shadow.

E.12 Counting Regions and Evaluation

By hand we count the number of regions identified as "ships" in the two images. We get 11 ships in the first image and and 21 in the second. The actual count appears to be 7 (plus 3 or 4 much smaller ones) in the first image and 17 in the second. The errors are caused by two factors: some single objects are split into two regions, and a few non-ships are identified as ships for one reason or another. In the first image one single ship was identified as a pair of ships because it was merged with one of the much smaller ships and part of the ship was not segmented with the rest because it had a much different intensity and no micro-edges. These two factors caused the length to width ratio to be more similiar to the pair of ships than to a single ship. A small section of water and a section of a pier were also identified as ships. Both of these regions resembled ships using length to width ratio, number of

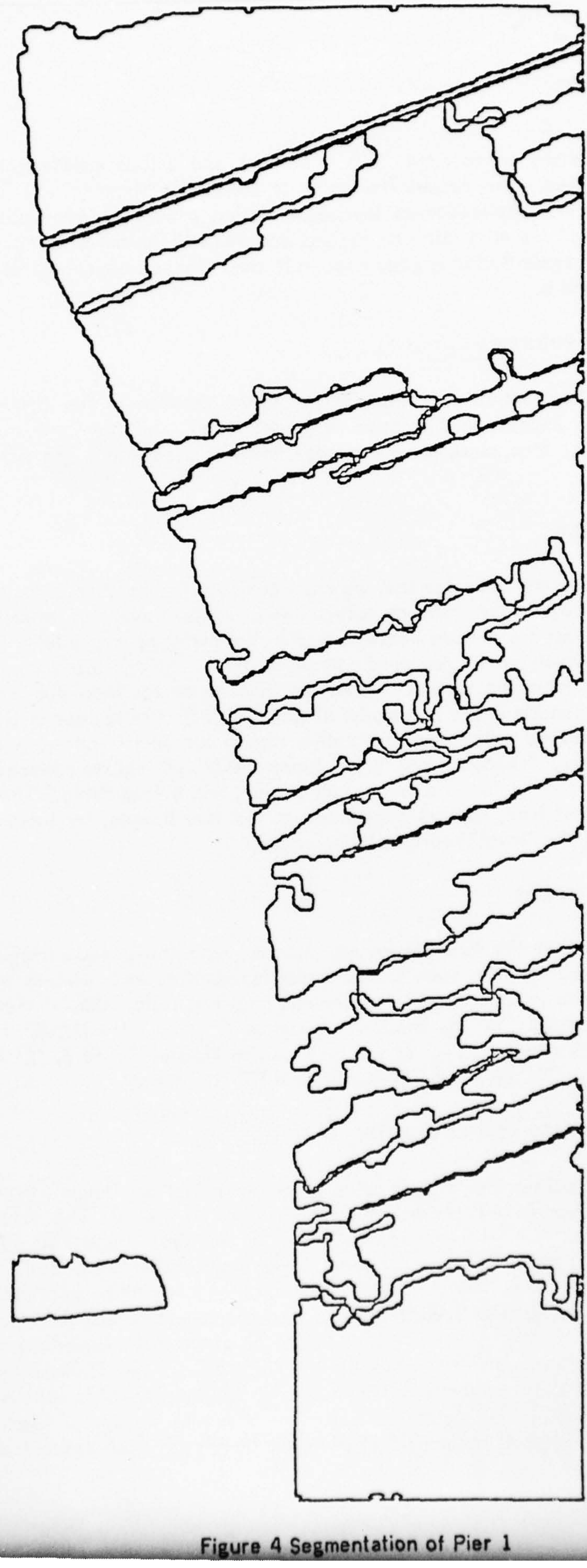
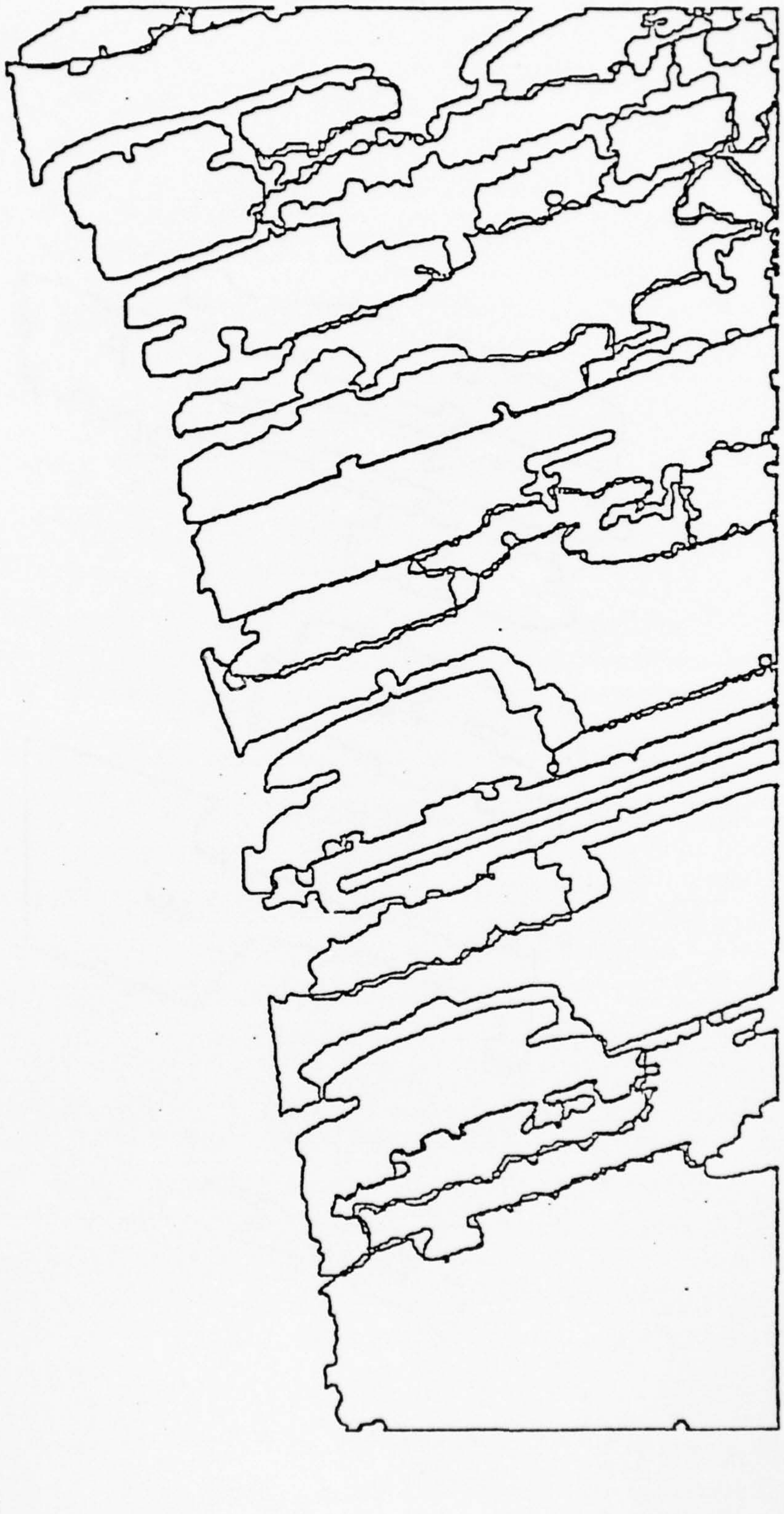


Figure 4 Segmentation of Pier 1



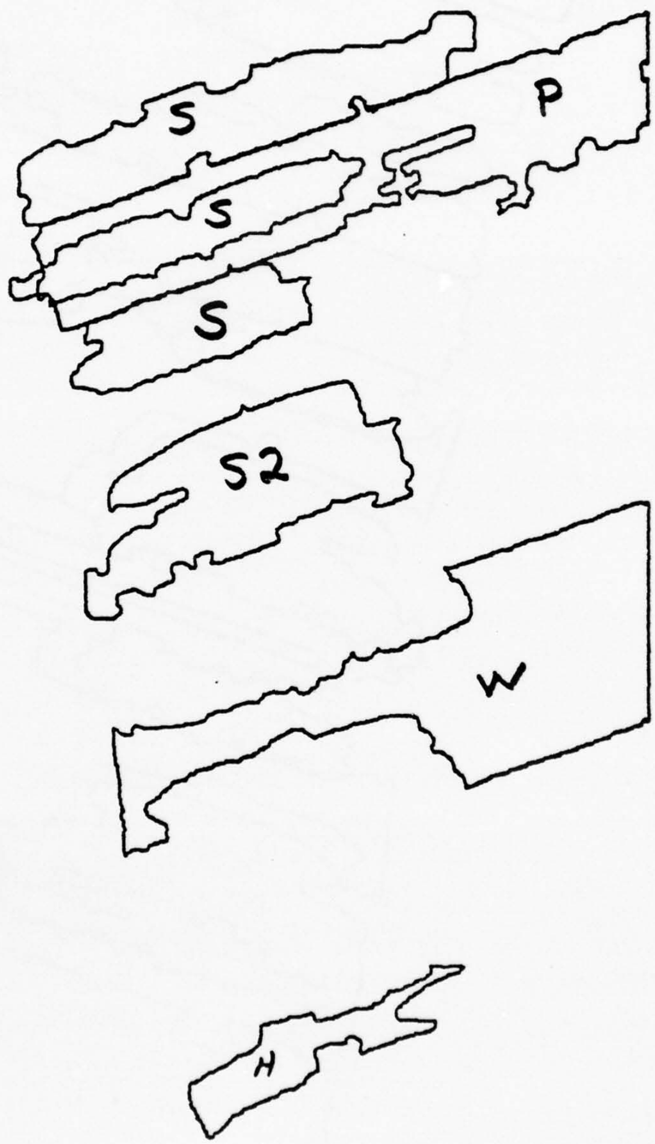


Figure 6 Regions Used for Pseudo Image

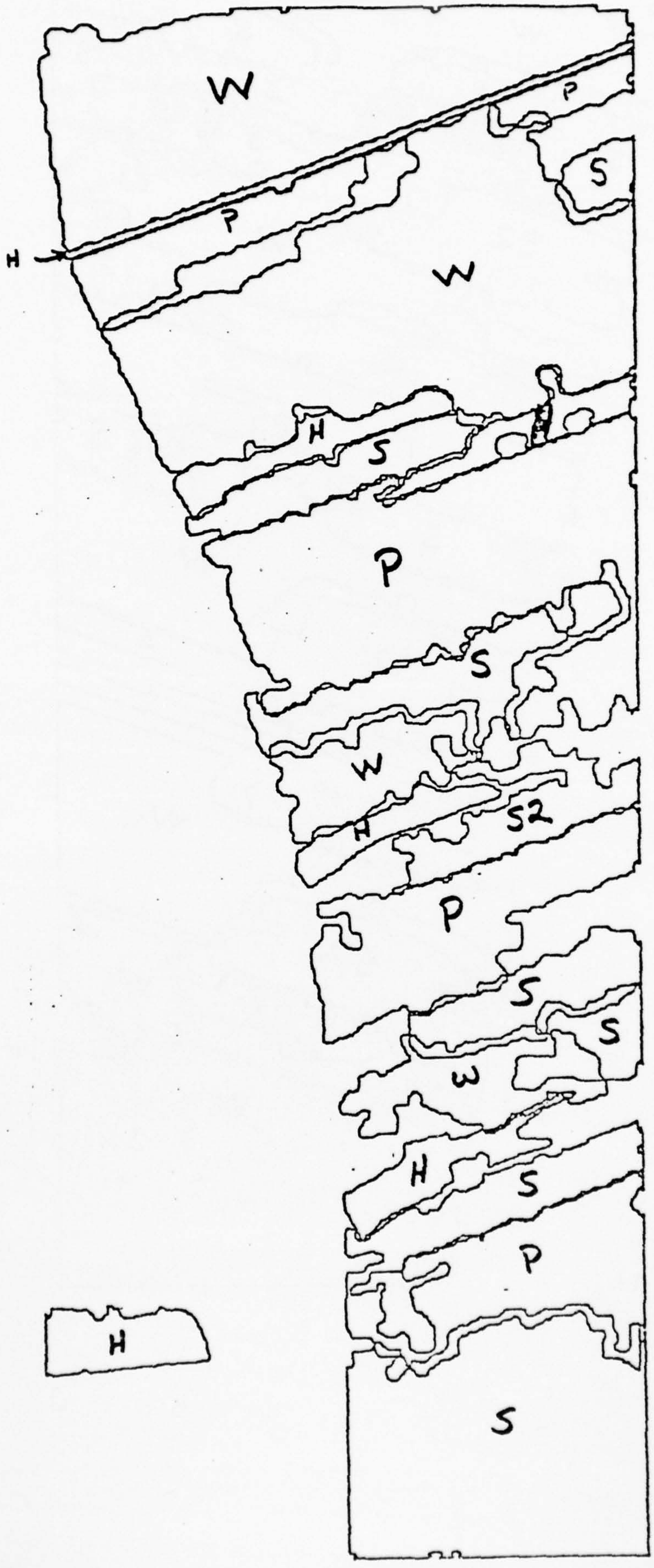


Figure 7 Matching for Pier 1

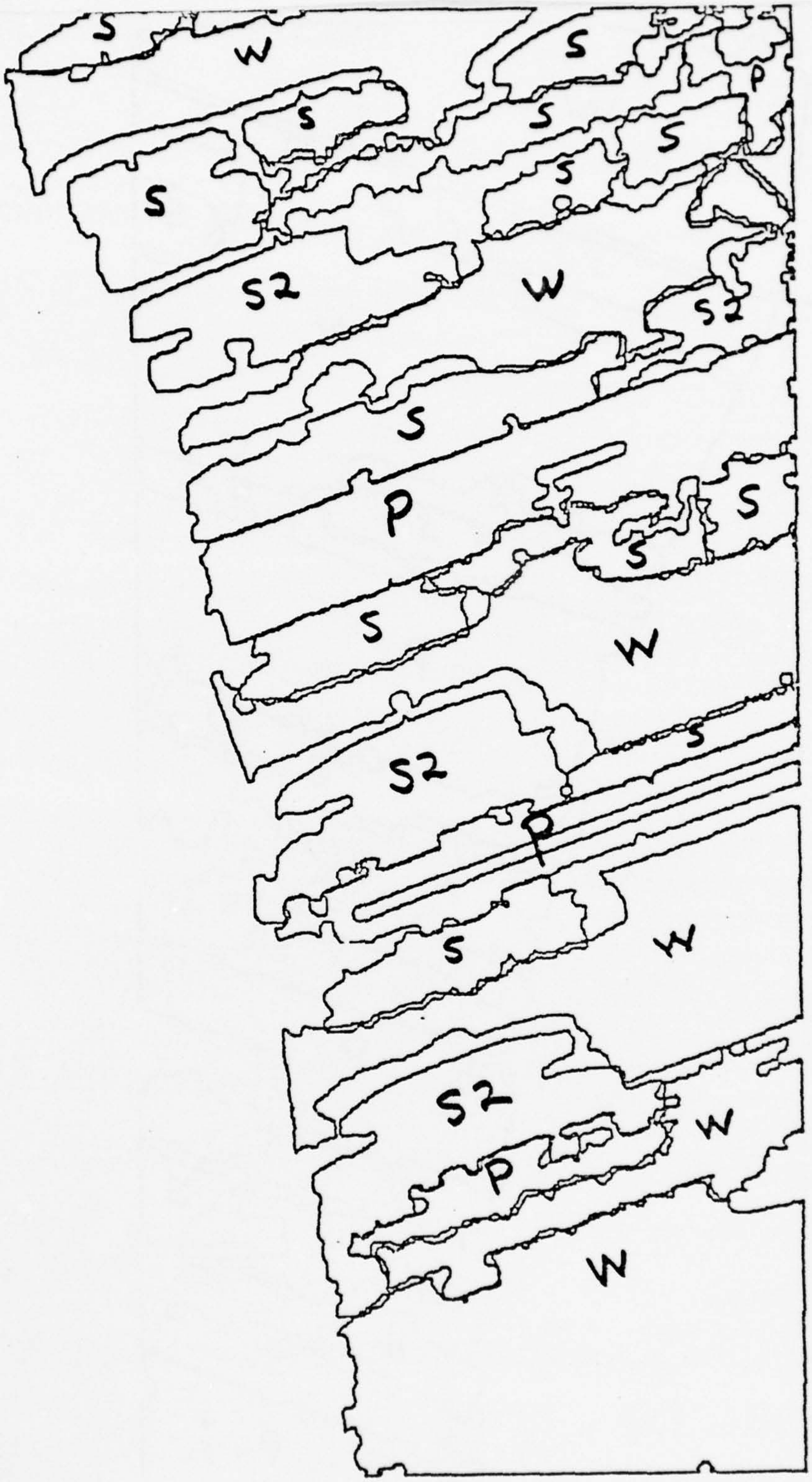


Figure 8 Matching for Pier 2

micro-edges, or intensity. Finally, a large block of water was indicated as a ship mostly because of the intensity and number of micro-edges. The use of size in matching these regions could possibly have eliminated this region as a ship, but the use of size would introduce other errors in the identification of other water regions or pier regions. In the second image, the errors are caused primarily by single objects being broken into two regions. Additionally, two pier sections are identified as ships and one small portion of a ship was not segmented. Two small (adjacent) ships were segmented as one region, but included some of the surrounding area and were indicated as a pair of ships (which is correct, but not really for the right reasons).

Appendix F Matching Results

This appendix will present the results of matching several regions from each of the scenes. For each scene we will present a set of corresponding regions with listings giving the contribution to the match rating for each of the features that were used. We will also give the mean and standard deviation of the match rating for the best match that was located (for all regions in the scene) and the same for the second best match. These two sets of numbers are given as an indicator of which features were important for all the regions in the image. We will also indicate the strengths which were applied to the features in the summary of the matching for the scene. A match rating of 0.0 indicates a perfect match (i.e. no differences between the regions). The rating of 0.0 for neighbors and relative position means that the regions matched using these features. For neighbors, it could also mean that there were no neighbors for the region being matched. The features are presented in the same order as they are computed by the machine. The features which are expected to remain constant, the strongest features, are presented first followed by the features which may change, the less strong features.

<u>The HOUSE scene matching from image 1 to image 2.</u>

Matching for Region "F"		Matching for Region "G"	
Size	-9.7800000	Size	-22.5300000
Color	-342.6464000	Color	-221.2416100
I Location	-34.5000000	I Location	-36.5000020
J Location	-9.1999969	J Location	-30.7999990
P^2/Area	-37.0892330	P^2/Area	-12.1112560
Neighbors	.0000000	Neighbors	.0000000
Orientation	-57.2462040	Orientation	-12.7095410
Relative Position	.0000000	Relative Position	.0000000
Length to Width	-11.1677210	Length to Width	-5.2967796
Fractional Fill	-17.8591000	Fractional Fill	-83.6862030
Match score:	-519.4886600	Match score:	-424.8753900

Matching for Region "H"		Matching for Region "A"	
Size	-3.7300000	Size	-100.3107400
Color	-257.2022600	Color	-297.7749500
I Location	-32.1000020	I Location	-18.4000020
J Location	-22.0000000	J Location	-26.2999990
P^2/Area	-12.5576780	P^2/Area	-44.2765010
Neighbors	.0000000	Neighbors	.0000000
Orientation	.0000000	Orientation	-18.7599610
Relative Position	.0000000	Relative Position	.0000000
Length to Width	-7.7273598	Length to Width	-7.9638023
Fractional Fill	-31.2788280	Fractional Fill	-66.8738250
Match score:	-366.5961200	Match score:	-580.6596800

Statistics Summary Best Match for House Scene

Feature	Mean	Stdv
Size	-34.3885950	33.2336810
Color	-221.7073300	116.1910900
I Location	-42.1375020	12.9814200
J Location	-18.2687500	13.8040630
P^2/Area	-45.4489740	43.2472150
Neighbors	-6.2500000	24.2061460
Orientation	-44.1049450	40.3305720
Relative Position	.0000000	.0000000
Length to Width	-14.4731270	16.1513830
Fractional Fill	-85.3697020	69.8006400
Match	-512.1489200	144.1054600

Statistics Summary Second Best Match for House Scene

Feature	Mean	Stdv
Size	-104.5731800	165.1729000
Color	-270.1108700	170.4281300
I Location	-87.6461530	54.8099320
J Location	-165.3846200	107.0603100
P^2/Area	-71.9028890	79.5340560
Neighbors	-38.4615380	48.6504260
Orientation	-49.6579920	49.6865540
Relative Position	.0000000	.0000000
Length to Width	-41.3859890	30.3387650
Fractional Fill	-212.9435800	144.6680400
Match	-1042.0668000	373.2167000

The house matching used all features with the same strength (the middle strength: 100). The nine color features (red, green, blue, density, hue, saturation, Y, I and Q) are the primary source of the low total match score. These color features matched almost as well for the second best matches (i.e. the wall areas have the same color properties so that these match as well for the second best match as for the best match). The shape and location features contributed the most to the selection of the correct match over the second best match.

The CITYSCAPE scene matching for image 1 to 2.

Matching for Region "A"		Region for Region "H"	
Size	-165.3300000	Size	-68.8681760
Color	-128.2613500	Color	-14.2949880
I Location	-10.4000020	I Location	-2.0999985
J Location	-131.6000000	J Location	-24.9000000
P^2/Area	-174.4339300	P^2/Area	-147.3116700
Neighbors	.0000000	Neighbors	.0000000
Orientation	-5.3301620	Orientation	-11.1646610
Relative Position	.0000000	Relative Position	.0000000
Length to Width	-26.9853900	Length to Width	-18.4006230
Fractional Fill	-176.1321900	Fractional Fill	-45.9115070
Match score:	-818.4730200	Match score:	-332.9516200

Matching for Region "Q"		Matching for Region "F"	
Size	-8.3900000	Size	-45.3200000
Color	-85.8519930	Color	-408.3855200
I Location	-37.1000000	I Location	-13.6000020
J Location	-2.3000031	J Location	-93.6000020
P^2/Area	-68.0003970	P^2/Area	-27.0875240
Neighbors	.0000000	Neighbors	.0000000
Orientation	-10.6830410	Orientation	-238.1794200
Relative Position	.0000000	Relative Position	.0000000
Length to Width	-14.6863860	Length to Width	-42.6815030
Fractional Fill	-29.5162720	Fractional Fill	-400.4948700
Match score:	-256.5280900	Match score:	-1269.3488000

Statistics Summary Best Match for Cityscape Scene

Feature	Mean	Stdv
Size	-75.3357540	68.2402880
Color	-126.1703900	106.4454900
I Location	-28.5047630	26.6346610
J Location	-33.9380970	34.3095070
P^2/Area	-67.2637510	55.0590450
Neighbors	-28.5714290	45.1753950
Orientation	-52.3166390	61.1797910
Relative Position	.0000000	.0000000
Length to Width	-32.4361080	23.5747520
Fractional Fill	-126.9055400	95.3435350
Match	-571.4424700	238.9614900

Statistics Summary Second Best Match for Cityscape Scene

Feature	Mean	Stdv
Size	-117.9716700	106.8301900
Color	-274.5621700	213.9505000
I Location	-91.9333350	71.4724930
J Location	-106.3476200	81.6396580
P^2/Area	-78.4391270	40.5079460
Neighbors	-23.8095240	42.5917710
Orientation	-62.4560520	75.7932040
Relative Position	.0000000	.0000000
Length to Width	-26.3227800	23.0091390
Fractional Fill	-257.3681700	397.7447500
Match	-1039.2104000	518.5977300

The cityscape also used all the available features at the same strengths. Region "F" moves to the left with respect to the other regions in the image so that the matching for the J location is lower. The match for this region was not very good, but it was the best available match. Size, color and position provided the best differentiation with the second best matches.

The LANDSAT scene matching results for image 1 to 2.

Matching for Region "A"		Matching for Region "B"	
Size	-1.7601245	Size	-22.9787500
I Location	-13.8541870	I Location	-22.9459230
J Location	-13.9665110	J Location	-8.6821089
P^2/Area	-17.2155220	P^2/Area	-4.1593509
Neighbors	.0000000	Neighbors	.0000000
Orientation	-5.7014599	Orientation	-5.2198796
Relative Position	.0000000	Relative Position	.0000000
Length to Width	-2.8751001	Length to Width	-10.2830400
Fractional Fill	-8.5236168	Fractional Fill	-102.1346500
Match score	-63.8965220	Match score	-176.4037000

Matching for Region "C"		Matching for Region "D"	
Size	.0000000	Size	-.0899541
I Location	-.0245972	I Location	-2.0444031
J Location	-.6405792	J Location	-.6666565
P^2/Area	-4.6169637	P^2/Area	-1.2948230
Neighbors	.0000000	Neighbors	.0000000
Orientation	-.3972000	Orientation	-2.8977603
Relative Position	.0000000	Relative Position	.0000000
Length to Width	-1.8653203	Length to Width	-.7425401
Fractional Fill	-6.0832943	Fractional Fill	-4.1045582
Match score	-13.6279550	Match score	-11.8406950

Matching for Region "E"		Matching for Region "G"	
Size	-.1248207	Size	-222.6387000
I Location	-12.6033170	I Location	-24.6552430
J Location	-15.9610330	J Location	-10.1603700
P^2/Area	-1.9656205	P^2/Area	-95.3620490
Neighbors	.0000000	Neighbors	.0000000
Orientation	-11.2387400	Orientation	-8.8417816
Relative Position	.0000000	Relative Position	.0000000
Length to Width	-2.1496801	Length to Width	-15.4305420
Fractional Fill	-10.0914540	Fractional Fill	-145.0512600
Match score	-54.1346650	Match score	-522.1399500

Statistics Summary Best Match for LANDSAT Scene

Feature	Mean	Stdv
Size	-35.4274720	76.8240620
I Location	-71.4245820	144.1344300
J Location	-142.6348500	328.9845300
P^2/Area	-27.6992470	35.5838790
Neighbors	.0000000	.0000000
Orientation	-8.1080639	6.7299419
Relative Position	.0000000	.0000000
Length to Width	-8.4677299	8.7104111
Fractional Fill	-39.7900170	54.2789580
Match	-333.5519600	501.5407000

Statistics Summary	Second Best Match for LANDSAT Scene	
Feature	*Mean*	Stdv
Size	-121.3912100	171.3832900
I Location	-250.5458500	274.3067900
J Location	-283.5227800	352.7298300
P^2/Area	-109.7119600	52.3837630
Neighbors	.0000000	.0000000
Orientation	-123.1217200	77.5954260
Relative Position	-14.2857140	34.9927110
Length to Width	-35.7855570	24.6940290
Fractional Fill	-217.3908900	94.9321610
Match	-1155.7557000	554.0008500

For the LANDSAT scene the matches for the correct regions were all very good. The one region not given here ("F") was mentioned in Chapter 6 as being matched to an incorrect region since there was no correct match possible. The snow region ("G") also has a low match rating, but this is due to the great change in the size of the regions, and the resulting shape and location changes. All of the features were used at the same strengths. Color was not used as a feature since it was specifically used in the segmentation process to select dark or bright regions. Size distinguishes the snow and lake regions as well as color does. The location was the most valuable feature in the matching of these regions, especially since they are all in a constant position with respect to each other.

<u>The SLR scene matching for image 1 to image 2.</u>

Matching for Region "A"	
Size	-421.0225800
I Location	-135.3566600
J Location	-101.6395800
P^2/Area	-1830772700
Neighbors	.0000000
Orientation	-4.2557373
Relative Position	.0000000
Length to Width	-52.1186600
Fractional Fill	-280.1908100
Color	-2.9404264
Match score:	-1180.6077000

Matching for Region "B"	
Size	-141.6448400
I Location	-82.8993030
J Location	-39.6789260
P^2/Area	-31.2576940
Neighbors	.0000000
Orientation	-3.3777008
Relative Position	.0000000
Length to Width	-67.1741600
Fractional Fill	-160.4823900
Color	-5.0000000
Match score:	-531.5150100

Matching for Region "F"	
Size	-42.3886270
I Location	-274.7080700
J Location	-45.9476170
P^2/Area	-5.5162125
Neighbors	.0000000
Orientation	-53.8893590
Relative Position	.0000000
Length to Width	-1.4643211
Fractional Fill	-25.4345970
Color	-.5063286
Match score:	-449.8551300

Matching for Region "E"	
Size	-8.6065234
I Location	-15.3521730
J Location	-6.9457397
P^2/Area	-23.0626020
Neighbors	.0000000
Orientation	-52.2719800
Relative Position	.0000000
Length to Width	-5.4783401
Fractional Fill	-13.0905160
Color	-2.0588236
Match score:	-126.8667000

Matching for Region "C"	
Size	-.0000006
I Location	-.2741394
J Location	-.7147675
P^2/Area	-22.8144140
Neighbors	.0000000
Orientation	.0000000
Relative Position	.0000000
Length to Width	-5.9841797
Fractional Fill	-3.0839002
Color	-2.8160920
Match score:	-35.6874940

Statistics Summary Best Match for SLR Scene

Feature	Mean	Stdv
Size	-103.5517100	149.9527100
I Location	-87.1394840	96.2091770
J Location	-34.2455310	34.4955330
P^2/Area	-64.6986730	65.1396250
Neighbors	.0000000	.0000000
Orientation	-27.2991290	24.8142710
Relative Position	.0000000	.0000000
Length to Width	-29.9410100	26.3539000
Fractional Fill	-86.8015170	101.1324100
Color	-2.3541285	1.4994871
Match	-436.0311900	374.2558600

Statistics Summary Second Best Match for SLR Scene

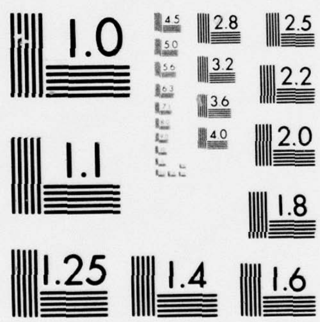
Feature	Mean	Stdv
Size	-120.4501900	159.1646000
I Location	-251.1913500	231.0709300
J Location	-293.4598500	186.9096100
P^2/Area	-68.6820400	51.4765780
Neighbors	.0000000	.0000000
Orientation	-47.4598220	47.7593780
Relative Position	.0000000	.0000000
Length to Width	-55.2706970	22.5380680
Fractional Fill	-95.9982940	97.2952800
Color	-1.9422436	1.8321452
Match	-934.4545000	402.5304900

The SLR matching used all but the color at the normal strength. Color was used as a variable feature (i.e. at a lower strength) since the intensity of the two images is very different. Location was the most important feature for distinguishing between the correct and incorrect matches. Region "C" was used as the original match to determine the location differences between the regions in the image. This means that the runway regions (e.g. "A") match best even though there are major differences in the size of the region. The orientation also played an important in the matches for this scene.

AD-A038 843 CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER --ETC F/G 15/4
CHANGE DETECTION AND ANALYSIS IN MULTI-SPECTRAL IMAGES,(U)
DEC 77 K E PRICE F44620-73-C-0074
UNCLASSIFIED AFOSR-TR-77-0448 NL

3 OF 3
AD
A038843

END
DATE
FILMED
5-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

The RURAL scene matching for image 2 to 3.

Matching for Region "A"		Matching for Region "W"	
Size	-19.7494980	Size	-1.1800000
Color	-21.6216220	Color	-18.6046510
P^2/Area	-12.9768590	P^2/Area	-35.8504200
Neighbors	.0000000	Neighbors	.0000000
Length to Width	-.2828002	Length to Width	-2.8739200
I Location	-.7900000	I Location	-9.6200008
J Location	-20.1700000	J Location	-10.2700000
Orientation	-4.2629900	Orientation	-1.5095444
Relative Position	.0000000	Relative Position	.0000000
Fractional Fill	-1.3166285	Fractional Fill	-5.5321140
Match score:	-81.1703980	Match score:	-85.4406500

Matching for Region "V"		Matching for Region "S"	
Size	-.4800000	Size	-.1700000
Color	-3.7500000	Color	-30.2325580
P^2/Area	-6.1597204	P^2/Area	-1.3160899
Neighbors	.0000000	Neighbors	.0000000
Length to Width	-1.5052394	Length to Width	-10.7532990
I Location	-11.7100010	I Location	-13.0000000
J Location	-10.6900000	J Location	-4.2500000
Orientation	-4.4164562	Orientation	-2.6876140
Relative Position	.0000000	Relative Position	.0000000
Fractional Fill	-2.3529410	Fractional Fill	-1.7819705
Match score:	-41.0643570	Match score:	-64.1915320

Statistics Summary Best Match for Rural Scene

Feature	Mean	Stdv
Size	-45.2715820	89.3841870
Color	-19.5820310	27.2795930
P^2/Area	-23.6892810	23.8833830
Neighbors	.0000000	.0000000
Length to Width	-10.7747260	10.9052340
I Location	-13.9888460	12.9805880
J Location	-7.9146155	7.6746485
Orientation	-7.2947964	7.5940160
Relative Position	.0000000	.0000000
Fractional Fill	-7.0943727	7.6343394
Match	-135.6102500	129.8113100

Statistics Summary Second Best Match for Rural Scene

Feature	Mean	Stdv
Size	-69.3681940	129.7705400
Color	-41.3515020	42.9294250
P^2/Area	-31.5566370	34.2044010
Neighbors	.0000000	.0000000
Length to Width	-20.2510250	18.8848660
I Location	-34.2234620	29.4807780
J Location	-19.8642310	29.1762240
Orientation	-12.3027760	8.6010383
Relative Position	-1.1538462	3.1948553
Fractional Fill	-9.6509346	8.6894248
Match	-239.7226100	200.4489700

The rural scene had a rotation difference between the two images. The first four features (size, color, P^2/A, neighbors, and length to width ratio) were weighted as constant features, the others were given less strength in the matching because of the rotation difference between the two images. The rural scene matching shows the matching for one of the large untextured regions ("A") where most of the features match reasonably well. The other regions are all small bright regions. Regions "V" and "W" are near the center and "S" is along the right side. For all of these the matches are very good. The shape features match well, but the low strength location and orientation features do not.

The URBAN-INDUSTRIAL scene matching for image 1 to 2

Matching for Region "B"		Matching for Region "M"	
Size	-11.6998260	Size	-.0000006
Color	-39.9014780	Color	-110.9890100
I Location	-7.5341187	I Location	-.3095856
J Location	-19.9582220	J Location	-.4444580
P^2/Area	-8.7344503	P^2/Area	-16.9599130
Neighbors	.0000000	Neighbors	.0000000
Orientation	-2.5670586	Orientation	.0000000
Relative Position	.0000000	Relative Position	.0000000
Length to Width	-.7706795	Length to Width	-6.5551395
Fractional Fill	-12.4246910	Fractional Fill	-10.8670520
Match score:	-103.5905200	Match score:	-146.1251600

Matching for Region "A"		Matching for Region "C"	
Size	-148.0667300	Size	-13.2795030
Color	-123.0337100	Color	-151.9607800
I Location	-22.7176090	I Location	-4.4280472
J Location	-82.8914220	J Location	-2.3690796
P^2/Area	-21.7353590	P^2/Area	-46.3663290
Neighbors	.0000000	Neighbors	.0000000
Orientation	-4.3826981	Orientation	-50.0000000
Relative Position	.0000000	Relative Position	.0000000
Length to Width	-7.2544403	Length to Width	-52.4477420
Color	-123.0337100	Color	-151.9607800
Fractional Fill	-47.3905450	Fractional Fill	-.4537506
Match score:	-457.4725100	Match score:	-321.3052400

Statistics Summary Best Match for Urban Scene

Feature	Mean	Stdv
Size	-19.8568030	36.3632550
Color	-106.1300000	63.2173490
I Location	-8.3780844	8.2233434
J Location	-18.1939820	24.6458770
P^2/Area	-33.8787570	27.4214310
Neighbors	.0000000	.0000000
Orientation	-22.5485200	22.5471100
Relative Position	.0000000	.0000000
Length to Width	-22.4529780	22.5962740
Fractional Fill	-12.2066080	11.3541210
Match	-243.6457300	108.3699000

Statistics Summary	Second Best Match for Urban Scene	
Feature	Mean	Stdv
Size	-16.9348940	20.7795830
Color	-121.5953200	111.7683300
I Location	-114.8360800	83.7923660
J Location	-103.3612100	70.7021240
P^2/Area	-50.7243920	50.6821050
Neighbors	.0000000	.0000000
Orientation	-23.2675560	21.7569930
Relative Position	-14.2857140	34.9927110
Length to Width	-20.5099130	16.1759810
Fractional Fill	-15.6398200	5.0961429
Match	-481.1548900	188.4103100

The urban-industrial scene matching was performed on the bright regions of the scene. The match results for region "M" and "B" were used for the size and locations adjustments which were then used for all the other matches. The location feature is the most important feature to distinguish the best match from the second best match. After the adjustments are made for location and size all the features are given the same strength.

The following two pier subsection matches are for matching the pier image with the model image of the pier area.
Several of the regions will match perfectly since these regions were used in the model of the pier area.

<u>The first PIER subscene matching to the model of a pier area.</u>

Matching for Region "W"		Matching for Region "W"	
Color	-39.9825780	Color	-3.2000000
P^2/Area	-124.8725100	P^2/Area	-102.5713400
Neighbors	.0000000	Neighbors	.0000000
Orientation	-8.2363777	Orientation	-28.8925800
Length to Width	-62.4860190	Length to Width	-14.2803610
Match score:	-235.5774900	Match score:	-148.9442800

Matching for Region "S"		Matching for Region "ZS"	
Color	.0000000	Color	-26.0416670
P^2/Area	.0000000	P^2/Area	-117.0344300
Neighbors	.0000000	Neighbors	.0000000
Orientation	.0000000	Orientation	-41.3927000
Length to Width	.0000000	Length to Width	-24.1059400
Match score:	.0000000	Match score:	-208.5747400

Statistics Summary	Best Match for First Pier Subscene	
Feature	Mean	Stdv
Color	-36.1623490	27.2753810
P^2/Area	-89.9218270	66.2324580
Neighbors	-28.5714290	45.1753950
Orientation	-32.4698720	33.6073110
Length to Width	-36.0700980	31.2858510
Match	-223.1955800	111.1476600

Statistics Summary	Second Best Match for First Pier Subscene	
Feature	Mean	Stdv
Color	-138.8544200	126.1953700
P^2/Area	-108.4028200	77.1022110
Neighbors	-9.5238096	29.3543520
Orientation	-33.3731190	32.7078730
Length to Width	-36.8562180	31.1277280
Match	-327.0703800	170.8690300

This matching for the first pier subsection shows three correct matches. Both of the illustrated water regions ("W") are correct, and the single ship match ("S") is also correct. the pair of ships match ("2S") is incorrect. This ship is adjacent to a very small ship and the two were segmented together, thus the length to width ratio and orientation are more like a pair of ships than like one ship. All the features that were used are given the same weight.

The second PIER subscene matching to the model of the pier area.

Matching for Region "W"	
Color	.0000000
P^2/Area	.0000000
Neighbors	.0000000
Orientation	.0000000
Length to Width	.0000000
Match score:	.0000000

Matching for Region "2S"	
Color	-13.4981270
P^2/Area	-59.7110540
Neighbors	.0000000
Orientation	-10.1687220
Length to Width	-17.3199610
Match score:	-100.6978600

Matching for Region "2S"	
Color	.0000000
P^2/Area	.0000000
Neighbors	.0000000
Orientation	.0000000
Length to Width	.0000000
Match score:	.0000000

Matching for Region "S"	
Color	-5.0656661
P^2/Area	-36.9228560
Neighbors	-100.0000000
Orientation	-2.4358387
Length to Width	-69.6544400
Match score:	-214.0788000

Matching for Region "S"	
Color	-2.2304833
P^2/Area	-2.5331585
Neighbors	.0000000
Orientation	-11.9736400
Length to Width	-4.5831203
Match score:	-21.3204020

Matching for Region "S"	
Color	-25.0000000
P^2/Area	-26.6103790
Neighbors	.0000000
Orientation	-9.0973797
Length to Width	-4.7275000
Match score:	-65.4352590

Statistics Summary	Best Match for Second Pier Subscene	
Feature	Mean	Stdv
Color	-34.7668240	38.3594060
P^2/Area	-49.3882260	50.0038690
Neighbors	-3.7037037	18.8852570
Orientation	-27.5113800	43.5144820
Length to Width	-18.6295520	23.6374160
Match	-133.9998900	111.4088600

Statistics Summary	Second Best Match for Second Pier Subscene	
Feature	Mean	Stdv
Color	-82.6211790	61.1719280
P^2/Area	-93.2314410	71.2189240
Neighbors	-3.7037037	18.8852570
Orientation	-32.7885530	43.7997250
Length to Width	-23.7676930	23.7880450
Match	-236.1125700	135.0442800

The matching results shown here are for several correct matches and one incorrect match, plus one match that is correct, but with a bad segmentation. The first single ship ("S") region (the fourth in this group) is really a small section of one of the piers, but this region matched well with the length to width ratio and with color (i.e. number of micro edges) so that it appears to be a ship. The last single ship region (the last region match given above) is really half a ship merged with part of a pier. This region did not match very well, but matched to the single ship better than to any other region. Two of the matches were perfect because these were the regions selected for the generation of the model of the pier region.

Appendix G Change Results

This appendix will present the results of performing a match on the previously located corresponding regions, with all features of the same strength. See Chapter 6 for a description of the matching procedure which is being used. This match operation will indicate where changes in feature values may have occurred. A change in a feature value will be indicated by a low rating for the feature to feature match for the region. A feature match rating of greater than -50.0 is considered to be a close match. We will give the matching for various regions in each scene and, for some of the features, we will indicate what kind of feature change caused the low rating (i.e. how much the feature changed). The color matches are a combination of all the color parameters available for that scene, such as the nine color parameters for the house and cityscape, and only one (intensity) for the monochromatic images. The matching region labels refer to the labels given the regions in the figures in the results section in Chapter 6.

<u>Change results for HOUSE image 1 to image 2.</u>

Region "H" the chimney

Size	-3.7300000	
Color	-298.5066100	Primarily the first image is brighter.
I Location	-32.1000020	
J Location	-22.0000000	
P^2/Area	-12.5576780	
Neighbors	.0000000	
Orientation	.0000000	Both are vertical
Relative Position	.0000000	
Length to Width	-7.7273598	
Fractional Fill	-31.2788280	
Match score:	-407.9004700	

The color parameters individually decrease by very little from the first image to the second: red by 8, green by 22, blue by 27, density by 19, and Y by 19; Q changes by about 2. The combination of these changes in several parameters causes a large change in the overall color feature.

Region "A", the sky

Size	-100.3107400	
Color	-329.1565300	Here the second image is brighter.
I Location	-18.4000020	
J Location	-26.2999990	
P^2/Area	-44.2765050	
Neighbors	.0000000	
Orientation	-18.7598650	
Relative Position	.0000000	
Length to Width	-7.9637985	
Fractional Fill	-66.8738250	
Match score:	-612.0412700	

Unlike the first region, in this case the second image is brighter, i.e. the features increase in value; red by 11, green by 6, blue by less than 1, density by 5, and Y by 7; Q changes by about 2. These smaller changes have more impact on the rating since the standard deviation of the feature is much smaller than for the first

region, so that the features are expected to change less between the images (see Chapter 6 on the computation of the feature to feature match).

Statistics Summary Feature	Best Match House Changes Mean	Stdv
Size	-34.3885950	33.2336810
Color	-257.2085000	136.7256500
I Location	-42.1375020	12.9814210
J Location	-18.2687500	13.8040630
P^2/Area	-45.4489750	43.2472150
Neighbors	-6.2500000	24.2081460
Orientation	-44.1049460	40.3305710
Relative Position	.0000000	.0000000
Length to Width	-14.4731260	16.1513830
Fractional Fill	-85.3697020	69.8006420
Match	-547.6501000	156.6535900

The color parameters change for all of the regions. The I location of the region changes because the camera was moved and no attempt was made to keep the objects in exactly the same place in the image. Size, P^2/Area, and the fractional fill changed in some regions because of minor segmentation changes, or because the region was on the edge of the image and was cut off, or, in the case of the bushes, some objects were segmented into two regions in one image rather than one region (corresponding to how they appear in the image).

<u>Change results for CITYSCAPE scene image 1 to image 2.</u>

Region "E" a building in the foreground		
Size	-155.7859800	The region is 1/3 larger in the second image.
Color	-164.8610600	Second image is darker.
I Location	-3.7000008	
J Location	-19.3000030	
P^2/Area	-147.8380300	Additional area due to segmentation differences.
Neighbors	.0000000	
Orientation	-50.0000000	Vertical in first, not so in the second.
Relative Position	.0000000	
Length to Width	-12.4128950	
Fractional Fill	-103.8498600	Due to differences in the segmentation.
Match score:	-657.7478300	

The differences indicated for this corresponding region are all due to the differences in the segmentation of the two images. A little additional area is included in this region in the second image which was not included in the region in the first image. This difference caused the changes in the size and shape parameters.

Region "F", a building in the background.

Size	-45.3200000	
Color	-482.4024400	
I Location	-13.5999980	
J Location	-93.5999980	Change in relative position (90 pixels to the left)
P^2/Area	-27.0875240	
Neighbors	.0000000	
Orientation	-238.1794200	Change due to occlusion in first image.
Relative Position	.0000000	
Length to Width	-42.6815030	
Fractional Fill	-400.4948800	Same as orientation.
Match score:	-1343.4258000	

This region had a change in its position relative to the other regions in the scene. This difference is indicated in the absolute position feature, but not the relative position feature because it is still above the same object. The position change also caused a change in how much of the object is occluded, which caused large differences in the orientation and fractional fill features.

Statistics Summary	Best Match Cityscape Changes	
Feature	Mean	Stdv
Size	-75.3357540	68.2402880
Color	-145.6638600	124.2838000
I Location	-28.5047620	26.6346620
J Location	-33.9380960	34.3095080
P^2/Area	-67.2637510	55.0590450
Neighbors	-28.5714290	45.1753950
Orientation	-52.3166390	61.1797910
Relative Position	.0000000	.0000000
Length to Width	-32.4361080	23.5747530
Fractional Fill	-126.9055400	95.3435380
Match	-590.9359300	253.5018500

Generally the changes in the cityscape scene are caused by the differences in the segmentations.

<u>Change results for LANDSAT scene image 1 to image 2.</u>

Region A, the large lake

Size	-1.7601245	
I Location	-13.8541870	Region is 19 pixels up in second image.
J Location	-13.9665110	Region is 17 pixels to the right in the second image.
P^2/Area	-17.2155220	
Neighbors	.0000000	
Orientation	-5.7014599	
Relative Position	.0000000	
Length to Width	-2.8751001	
Fractional Fill	-8.5236168	
Match score:	-63.8965220	

Only minor location changes are indicated. These location differences are adjusted by the location differences of other corresponding regions.

Region "F", incorrectly matched lake		
Size	-.3999541	
I Location	-423.8444100	
J Location	-948.3666800	
P^2/Area	-69.2803960	
Neighbors	.0000000	
Orientation	-22.4596250	
Relative Position	.0000000	
Length to Width	-25.9278870	
Fractional Fill	-2.5412903	
Match score:	-1492.8202000	

This lake is incorrectly matched, but there was no correct corresponding region. An analysis of the differences would indicate that this match is very unlikely to be correct since the location differences of a stationary object are large (400 and 900 pixels).

Region "G", the snow region		
Size	-222.6387000	This is the desired difference, 272861 pixels.
I Location	-24.6552430	
J Location	-10.1603700	
P^2/Area	-95.3620490	The shape changes due to the size change.
Neighbors	.0000000	
Orientation	-8.8417816	
Relative Position	.0000000	
Length to Width	-15.4305420	
Fractional Fill	-145.0512600	Also due to the size change.
Match score:	-522.1399500	

The change in the size of this region were the desired results of the matching procedure. The size differences also cause the changes in the shape parameters.

Statistics Summary	Best Match LANDSAT Changes	
Feature	Mean	Stdv
Size	-35.4274720	76.8240620
I Location	-71.4245820	144.1344300
J Location	-142.6348500	328.9845300
P^2/Area	-27.6992470	35.5838790
Neighbors	.0000000	.0000000
Orientation	-8.1080639	6.7299419
Relative Position	.0000000	.0000000
Length to Width	-8.4677299	8.7104111
Fractional Fill	-39.7900170	54.2789580
Match	-333.5519600	501.5407000

The differences indicated for the scene as a whole are due to the incorrect match, and some to the snow cover changes.

Change results for SLR scene image 1 to image 2.

Region "A"		
Size	-421.0225800	First is 3 times the second.
Color	-29.4642870	
I Location	-135.3566600	Second is 130 pixels down
J Location	-101.6395800	Second is 100 pixels to right
P^2/Area	-183.0772700	
Neighbors	.0000000	
Orientation	-4.2557373	Both near -.3 radians.
Relative Position	.0000000	
Length to Width	-52.1186600	The first is wider, but not longer.
Fractional Fill	-280.1908200	
Match score:	-1207.1256000	

All the differences indicated here are due to the difference in segmentation. The region in the first image contains much more area than in the second image.

Region "C"	
Size	-.0000006
Color	-28.1609190
I Location	-.2741394
J Location	-.7147675
P^2/Area	-22.8144150
Neighbors	.0000000
Orientation	.0000000
Relative Position	.0000000
Length to Width	-5.9841795
Fractional Fill	-3.0839005
Match score:	-61.0323210

This region was used to adjust the later matching regions for location differences.

Statistics Summary	Best Match SLR Changes	
Feature	Mean	Stdv
Size	-103.5517100	149.9527100
Color	-23.5412910	14.9948700
I Location	-87.1394840	96.2091770
J Location	-34.2455310	34.4955330
P^2/Area	-64.6986730	65.1396250
Neighbors	.0000000	.0000000
Orientation	-27.2991290	24.8142710
Relative Position	.0000000	.0000000
Length to Width	-29.9410100	26.3539000
Fractional Fill	-86.8015180	101.1324100
Match	-457.2183400	377.6868500

The matches for "A" and "B" caused changes in the size, location, and shape features.

Change results for RURAL scene image 2 to image 3.

Region "A", large untextured region at the top.

Size	-19.7494980	
Color	-21.6216220	
I Location	-7.9000015	
J Location	-201.7000000	Region is second image is 200 pixels down.
P^2/Area	-12.9768600	
Neighbors	.0000000	
Orientation	-42.6299020	
Relative Position	.0000000	
Length to Width	-.2827988	
Fractional Fill	-13.1662830	
Match score:	-320.0269600	

The change in the J location is due to the rotation difference between the two images. For a region this large, the orientation is not really very meaningful so the change is not important.

Region "W", long, thin bright region in center.

Size	-1.1800000	
Color	-18.6046510	
I Location	-96.2000060	Region in second image is 100 pixels up.
J Location	-102.7000000	Region in second image is 100 pixels to right.
P^2/Area	-35.8504200	
Neighbors	.0000000	
Orientation	-15.0954420	Differs by .08 radians.
Relative Position	.0000000	
Length to Width	-2.8739204	
Fractional Fill	-55.3211440	
Match score:	-327.8255800	

The fractional fill difference is due to the rotation difference, and the fact that the region is long and thin so that it fills little of the MBR.

Region "V", small bright region near center.

Size	-.4800000	
Color	-3.7500000	
I Location	-117.1000100	
J Location	-106.8999900	
P^2/Area	-6.1597214	
Neighbors	.0000000	
Orientation	-44.1645640	Differs by .23 radians.
Relative Position	.0000000	
Length to Width	-1.5052376	
Fractional Fill	-23.5294110	
Match score:	-303.5889400	

Again, the differences are due to the rotation differences. This region could be used to adjust the orientation differences in other matches in this scene.

Region "Z", small bright region near center

Size	-.1800000	
Color	-5.8823529	
I Location	-116.5999900	
J Location	-43.5000010	
P^2/Area	-78.6380630	
Neighbors	.0000000	
Orientation	-50.4670010	Differs by .25 radians.
Relative Position	.0000000	
Length to Width	-22.4989620	
Fractional Fill	-16.3043480	
Match score:	-334.0707200	

The orientation difference is in the same direction as for region "V". This region is also elongated so that the orientation differences will produce a change in the other shape parameters.

Region "S", bright region along the right side.

Size	-.1700000	
Color	-30.2325580	
I Location	-130.0000000	
J Location	-42.5000000	
P^2/Area	-1.3160896	
Neighbors	.0000000	
Orientation	-26.8761410	Differs by .13 radians.
Relative Position	.0000000	
Length to Width	-10.7533000	
Fractional Fill	-17.8197080	
Match score:	-259.6678000	

Statistics Summary	Best Match Rural Changes	
Feature	Mean	Stdv
Size	-45.2715820	89.3841870
Color	-19.5820310	27.2795920
I Location	-139.8884600	129.8058800
J Location	-79.1461530	76.7464820
P^2/Area	-23.6892830	23.8833810
Neighbors	.0000000	.0000000
Orientation	-72.9479640	75.9401630
Relative Position	.0000000	.0000000
Length to Width	-10.7747260	10.9052330
Fractional Fill	-70.9437270	76.3433980
Match	-462.2439200	305.2541500

All the matches indicated some change in the absolute position since there is a rotational difference between the two images. Many of the size differences are due to differences in the segmentation of the large general untextured regions and not to any real change in the size. All the regions also had an orientation change caused by the rotation. If the orientation adjustment had been included, the orientation differences would have been much less for most of the small bright regions.

Change results for URBAN-INDUSTRIAL scene image 1 to image 2.

Region "A" left most region near the top

Size	-148.0667300	Region in the second image is 2.5 times as large
Color	-123.0337100	Region in second image is brighter by 22
I Location	-22.7176090	
J Location	-82.8914220	Change due to larger size.
P^2/Area	-21.7353590	
Neighbors	.0000000	
Orientation	-4.3826981	
Relative Position	.0000000	
Length to Width	-7.2544403	
Fractional Fill	-473.9054300	
Match score:	-883.9874000	

The region is brighter in the second image by more then 2 times the standard deviation of the average intensity of the region in the first image. There is an actual change in the size of the corresponding regions, i.e. a larger area is bright in the second image. The fractional fill change is due to the size change: the region gets wider but not longer.

Region "M" left most round region

Size	-.0000006	
Color	-110.9890100	Region in second image is brighter by 20
I Location	-.3095856	
J Location	-.4444580	
P^2/Area	-16.9599130	
Neighbors	.0000000	
Orientation	.0000000	
Relative Position	.0000000	Neither has a defined orientation.
Length to Width	-6.5551395	
Fractional Fill	-108.6705200	
Match score:	-243.9286300	

This matching pair was used to adjust the location and size for future matches (including this change indication match).

Region "E" topmost round region

Size	-13.5611500	Region in first image is not complete
Color	-120.6106900	Second image is brighter
I Location	-2.1140404	
J Location	-1.3789978	
P^2/Area	-68.2479860	Caused by the missing portion
Neighbors	.0000000	
Orientation	-50.0000000	Region in first has a defined orientation, second is round
Relative Position	.0000000	
Length to Width	-47.6596410	
Fractional Fill	-10.2315560	
Match score:	-313.8040600	

The region in the first image covers only a portion of the round object. The top part is lost due to the shadows which occur in the first image. This causes differences in the size and the shape features. As with all bright regions in this image, there is an intensity change.

Statistics Summary	Best Match Urban scene	
Feature	Mean	Stdv
Size	-19.8568030	36.3632550
Color	-106.1300000	63.2173480
I Location	-8.3780844	8.2233434
J Location	-18.1939820	24.6458770
P^2/Area	-33.8787570	27.4214310
Neighbors	.0000000	.0000000
Orientation	-22.5485200	22.5471100
Relative Position	.0000000	.0000000
Length to Width	-22.4529780	22.5962740
Fractional Fill	-122.0660800	113.5412100
Match	-353.5052000	173.5301100

All the regions were brighter in the second image than in the first image. The size and location were not significantly different between the two images because we were using the size and location adjustments calculated from the changes for region "M".